NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TOWARD A MORE RESPONSIVE CONSUMABLE MATERIEL SUPPLY CHAIN: LEVERAGING NEW METRICS TO IDENTIFY AND CLASSIFY ITEMS OF CONCERN

by

Andrew R. Haley

June 2016

Thesis Advisor:	Robert Koyak
Second Reader:	Peter Ward

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2016	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
TOWARD A MORE RESPONSIVE CONSUMABLE MATERIEL SUPPLY CHAIN: LEVERAGING NEW METRICS TO IDENTIFY AND CLASSIFY ITEMS OF CONCERN

5. FUNDING NUMBERS

6. AUTHOR(S) Andrew R. Haley

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943–5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Supply Systems Command
5450 Carlisle Pike
P.O. Box 2050
Mechanicsburg, PA 17055

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
2016-1

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

We develop a classification system for U.S. Navy consumable items to give the Naval Supply Systems Command (NAVSUP) a better position for advocacy regarding these assets. The Defense Logistics Agency (DLA) is responsible for the procurement, storage, and distribution of the Navy's consumable assets. Its inventory system is highly dynamic, and items may be requisitioned for long periods without undue delay followed by sudden, unexpected shortages that directly affect Navy combat readiness.

We propose a new metric, customer time limit (CTL), which normalizes the requisition fulfillment time according to priority level and the physical location of the customer. Using this metric, we essentially classify inventory items as problematical with respect to two different criteria: whether the median CTL exceeds a nominal threshold, and whether CTL exhibits an increasing trend. To apply this classification, nonparametric statistical methods are used based on consumable requisition data for calendar years 2013 through 2015, resulting in three categories: *NSNs at Risk*, *Bad Actors*, or *Bad Actors with Trend*.

Collectively, we find that *NSNs at Risk* and *Bad Actors with Trend* constitute approximately 1% in both U.S. Navy consumable item population and annual consumable expenditure ($19 million out of $1.9 billion purchased), and that *Bad Actors* comprise approximately 2% of U.S. Navy consumable item population and 7% of annual consumable expenditure ($140 million out of $1.9 billion purchased).

14. SUBJECT TERMS
Defense Logistics Agency (DLA), Naval Supply Systems Command (NAVSUP), logistics, inventory, consumable, NSNs at Risk, Bad Actors, Bad Actors with Trend, items of concern, customer time limit (CTL), coefficient of variation (CV), Spearman rank correlation test

15. NUMBER OF PAGES
103

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540–01–280–5500 Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

TOWARD A MORE RESPONSIVE CONSUMABLE MATERIEL SUPPLY CHAIN: LEVERAGING NEW METRICS TO IDENTIFY AND CLASSIFY ITEMS OF CONCERN

Andrew R. Haley
Lieutenant Commander, Supply Corps, United States Navy
B.S., University of Wisconsin-Madison, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2016

Approved by: Professor Robert Koyak
Thesis Advisor

CDR Peter Ward
Second Reader

Patricia Jacobs
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

We develop a classification system for U.S. Navy consumable items to give the Naval Supply Systems Command (NAVSUP) a better position for advocacy regarding these assets. The Defense Logistics Agency (DLA) is responsible for the procurement, storage, and distribution of the Navy's consumable assets. Its inventory system is highly dynamic, and items may be requisitioned for long periods without undue delay followed by sudden, unexpected shortages that directly affect Navy combat readiness.

We propose a new metric, customer time limit (CTL), which normalizes the requisition fulfillment time according to priority level and the physical location of the customer. Using this metric, we essentially classify inventory items as problematical with respect to two different criteria: whether the median CTL exceeds a nominal threshold, and whether CTL exhibits an increasing trend. To apply this classification, nonparametric statistical methods are used based on consumable requisition data for calendar years 2013 through 2015, resulting in three categories: *NSNs at Risk*, *Bad Actors*, or *Bad Actors with Trend*.

Collectively, we find that *NSNs at Risk* and *Bad Actors with Trend* constitute approximately 1% in both U.S. Navy consumable item population and annual consumable expenditure ($19 million out of $1.9 billion purchased), and that *Bad Actors* comprise approximately 2% of U.S. Navy consumable item population and 7% of annual consumable expenditure ($140 million out of $1.9 billion purchased).

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AAC	acquisition advice code
ACWT	average customer wait time
ALT	administrative lead time
CART	classification and regression trees
COG	cognizance symbol
CONUS	Continental United States
CTL	customer time limit
CV	coefficient of variation
CY	calendar year
DLA	Defense Logistics Agency
DOD	Department of Defense
EBS	Enterprise Business System
ERP	Enterprise Resource Planning
FAR	Federal Acquisition Regulations
FSC	Federal Supply Classification
FY	fiscal year
GAO	Government Accountability Office
IDIQ	indefinite delivery indefinite quantity
LCB	lower confidence bound
LCBs	lower confidence bounds
LTC	long-term contract
LTCs	long-term contracts
NAVSUP	Naval Supply Systems Command
NAVSUP WSS	Naval Supply Systems Command Weapons Systems Support
NSN	National Stock Number
PLT	production lead time
TYCOM	Type Commander
UMMIPS	Uniform Materiel Movement and Issue Priority System

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

In 2001, the Defense Logistics Agency (DLA) began to assume control of all consumable materiel from each military branch, creating a unified consumable military inventory. This change resulted in improving overall inventory system efficiency at the expense of individual service branch oversight (Diaz, Cardenas, & Brito, 2006). DLA is managing a highly dynamic inventory system, where demand may be extremely infrequent or erratic. To mitigate these effects, DLA has recently added two contractor-based proprietary resource planning systems, but their output is unable to be critically reviewed (GAO, 2014). Despite these new planning tools, intermittent and persistent shortages still exist, directly impact Naval combat readiness.

Naval Supply Systems Command (NAVSUP) is aware of these shortages and has internally labeled the items that experience shortages as "Bad Actors." However, prior to this thesis the term lacked an official definition. In this thesis, we develop a formal consumable inventory classification scheme and define three categories of items of concern: *NSNs at Risk*, *Bad Actors*, or *Bad Actors with Trend*.

The NAVSUP Inform-21 database is the source of data for this research, which represents an official record of all requisitions since the Navy transitioned to Enterprise Resource Planning (ERP) in 2010 (May, 2014). After filtering the data to reflect an appropriate scope of research, approximately 3 million requisitions remain. We execute original scripts, both in the Python and R languages, to accomplish our research objectives and analysis.

We argue that existing metrics, such as average customer wait time (ACWT), are insufficient to adequately describe items of concern. We propose a new metric, called customer time limit (CTL), that takes into account both the time to fulfill a requisition and the time allowance for that requisition depending on the priority level and geographic location of the customer as prescribed in NAVSUP Publication 485 in 2015. We also desire to incorporate a measure of demand variability into our analysis. The coefficient of variation (CV) is a statistical metric that is widely used in inventory management to

measure variability, and we apply it to our research in order to limit the scope of the items considered to a subset with CV scores that are considered "forecastable" as defined by Rigoni and Correia de Souza in 2016.

We use statistical modeling to relate CTL to a set of predictor variables from which residuals may be obtained for identifying items that warrant scrutiny. In order to build the most statistically significant models, the data is isolated by Federal Supply Classification (FSC) code. Given the limited scope of this thesis, we select three FSC codes to present in depth that have an important impact on Naval combat readiness: FSC code 5331 (O-Rings; containing approximately 149,000 requisitions), FSC code 4930 (Lubrication and Fuel Dispensing Equipment; containing roughly 6,000 requisitions), and FSC code 1285 (Fire Control Radar Equipment; containing approximately 600 requisitions). We build three separate regression trees on the basis of data within these FSC codes, and use the resulting residuals specific to each regression model to identify a statistical trend over time.

The specific method used to identify the statistical trend in the residuals is the Spearman rank correlation test. Non-parametric in origin, its results are tested against the null hypothesis that time and residuals have no association. Items that reject the null hypothesis are part of the definition of the two categories of items of concern that require a trend, *NSNs at Risk* and *Bad Actors with Trend*.

Our classification scheme also defines acceptable ranges for the median score calculated from each item's actual customer time limit (CTL) values. We analyze the range of CTL values during a particular year and calculate a 95% lower confidence bound (LCB) for the median CTL score per item via a non-parametric method first presented by Conover in 1999. Each category of item of concern defines its own particular acceptable lower and upper bound for LCB of the median CTL score. Combining criteria using 95% LCBs of the median CTL, Spearman rank correlation test results, and CV scores restricted to only "forecastable" items produces the formal classification scheme for *NSNs at Risk*, *Bad Actors*, and *Bad Actors with Trend* (see Table 1).

Table 1. Items of Concern: Categories and Associated Rules

Category	LCB of the Median CTL	Spearman Test Included
NSNs at Risk	80% to 99%	Yes
Bad Actors	at least 100%	No
Bad Actors with Trend	at least 100%	Yes

A CV score of less than 2, which is considered "forecastable," applies to all three categories.

Using the formal scheme from Table 1, items from within each of the three chosen FSC codes are modeled, classified, and results presented. Finally, we extend the analysis to each unique FSC code in sufficient depth to comprehend the aggregate impact of items of concern to the U.S. Navy. Collectively, we find that *NSNs at Risk* and *Bad Actors with Trend* constitute approximately 1% in both U.S. Navy consumable item population and annual consumable expenditure, and that *Bad Actors* comprise approximately 2% of U.S. Navy consumable item population and 7% of annual consumable expenditure (see Table 2).

Table 2. Items of Concern Summary Statistics, CY2015

Category	*NSNs At Risk*	*Bad Actors*	*Bad Actors with Trend*
U.S. Navy Consumable Population (unique NSNs)	268	6,128	657
U.S. Navy Consumable Population (%)	0.1%	2.0%	0.2%
Annual Consumable Expenditure ($, millions)	$3.8	$143.1	$19.4
Annual Consumable Expenditure (%)	0.2%	7.5%	1.0%
Total U.S. Navy Consumable Population (unique NSNs)	**300,281**		
Total Annual Consumable Expenditure ($, millions)	**$1,910**		

Items of Concern represent the collective group of consumable *NSNs at Risk*, *Bad Actors*, and *Bad Actors with Trend*. We analyze over 300 unique FSC codes in the data in sufficient depth to obtain basic summary statistics on each category.

Although small in percentage of total consumable population and amount spent, all three categories of items of concern have a potentially large impact on Naval readiness and warrant further scrutiny.

We conclude the thesis by offering three recommendations. First, replace average customer wait time (ACWT) with customer time limit (CTL) as the primary supply system metric for measuring responsiveness as a function of time. Second, analyze each unique FSC code in greater depth in order to refine the specific regression mode used, and continue to generate additional items of concern in the future via our original Python and R scripts. Finally, we recommend that NAVSUP should use our results as a basis for a dialogue with DLA to improve the inventory position of the wholesale consumable inventory system, and we discuss two existing methods available, procurement under long-term contracts (LTCs) and collaborative forecasting, to accomplish that goal.

References

Diaz, F. S., Cardenas, N., & Brito, L. O. (2006). *Implementing the National Inventory Management Strategy: A case study on DLA's National Inventory Management Strategy (NIMS)* (Master's thesis, Naval Postgraduate School, 2006). Monterey: Naval Postgraduate School. Retrieved from http://hdl.handle.net/10945/10066

May, B. (2014). *Applying ABC analysis to the Navy's Inventory Management System* (Master's thesis, Naval Postgraduate School, 2014). Monterey: Naval Postgraduate School. Retrieved from http://hdl.handle.net/10945/43953

United States Government Accountability Office Rep. No. GAO-14-495 (2014). Report to the Subcommittee on Readiness, Committee on Armed Services, House of Representatives.

ACKNOWLEDGMENTS

I would like to thank my wife, Allie Ai Haley, for enduring the long hours required for the Operations Research program, and to my daughter, Akira Haley, for being a bright shining star in our lives. This thesis journey began before Christmas 2015 when I visited Naval Supply Systems Command headquarters and was graciously hosted by Mr. Steve Weir, Commander Michael Benedetto, and Captain Eric Morgan. For the remainder of the process, Professors Robert Koyak and Geraldo Ferrer, Commander Peter Ward, and Ms. Marianne Taflinger all deserve special credit for generously offering their time, brainpower, and mentorship.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

For want of a nail the shoe was lost,
for want of a shoe the horse was lost;
and for want of a horse the rider was lost;
being overtaken and slain by the enemy,
all for want of care about a horse-shoe nail.

—Benjamin Franklin, *The Way to Wealth* (1758)

In the world of U.S. Navy logistics, two organizations are responsible for procurement, storage, and distribution of parts. Naval Supply Systems Command Weapon Systems Support (NAVSUP WSS) is responsible for repair parts, and the Defense Logistics Agency (DLA) is responsible for consumable parts. Repair parts typically are electronic suites or parts engineered with several layers of subcomponents, while consumable parts are generally the bit piece parts, such as screws, nails and washers. The Navy manages NAVSUP WSS and can change its policies and procedures at will. However, beginning in 2001, in accordance with the National Inventory Management Strategy, the Navy, along with the other military services, turned over responsibility for all consumable parts to DLA to create a single national inventory of consumable materiel (Diaz, Cardenas, & Brito, 2006). As a result of this change, the Navy gained efficiency but lost some level of oversight in its consumable supply chain, as DLA is an independent agency.

DLA is the federal government's largest logistics support agency, supporting all branches of the U.S. military and 110 foreign allies. DLA provides nearly ninety percent of the military's spare parts, supporting over 2,400 unique weapons systems. Its wholesale procurement is managed by Primary Field Level Activities, such as DLA Land and Maritime, responsible for surface ship and submarine parts, and DLA Aviation, responsible for aviation parts. DLA also has established inventory storage nodes in locations proximate to major fleet concentration areas (Defense Logistics Agency, 2016).

A. SUPPLY SYSTEM OVERVIEW

In the U.S. Navy supply system, there are essentially two levels of logistics, the wholesale level and retail level. Despite being managed by two different organizations, the essential structure is applicable to both consumable and repair parts.

1. U.S. Navy Wholesale Level

The wholesale level can be thought of as the "big picture." Managers at this level are responsible for maintaining the system as a whole. Either NAVSUP WSS or DLA monitors the overall inventory of each part, forecasts demand, creates contracts with private industry for replenishment, and maintains inventory nodes for wholesale storage (see Figure 1). For DLA, specific oversight of each part usually is managed by teams organized by the four-digit Federal Supply Classification (FSC) code, which classifies a part by type of materiel. When the consumable supply chain experiences a shortage of a part, there may be a variety of causes. Availability of materiel is influenced by the number of commercial suppliers available and the type of contract DLA may initiate with them. A highly variable demand pattern increases the difficulty of setting reasonable wholesale inventory levels, which may lead to shortages. Also, the amount of time required for a commercial supplier to manufacture the item, and the time required to meet technical specifications such as first article and production lot testing (K. J. Jackson, email to the author, 25 April 2016), influences the ability of the inventory system to respond quickly to shortages.

2. U.S. Navy Retail Level

The retail level of logistics is the "tactical" level, at which customers order on behalf of their units, and maintain local inventory sites not monitored by the wholesale system (see Figure 1). The customer may order materiel for direct turnover to a work center or for stock replenishment in local inventory. Managers at this level have limited visibility of the wholesale inventory status of materiel but attempt to make their critical needs known to their Type Commander (TYCOM) or to the wholesale inventory manager.

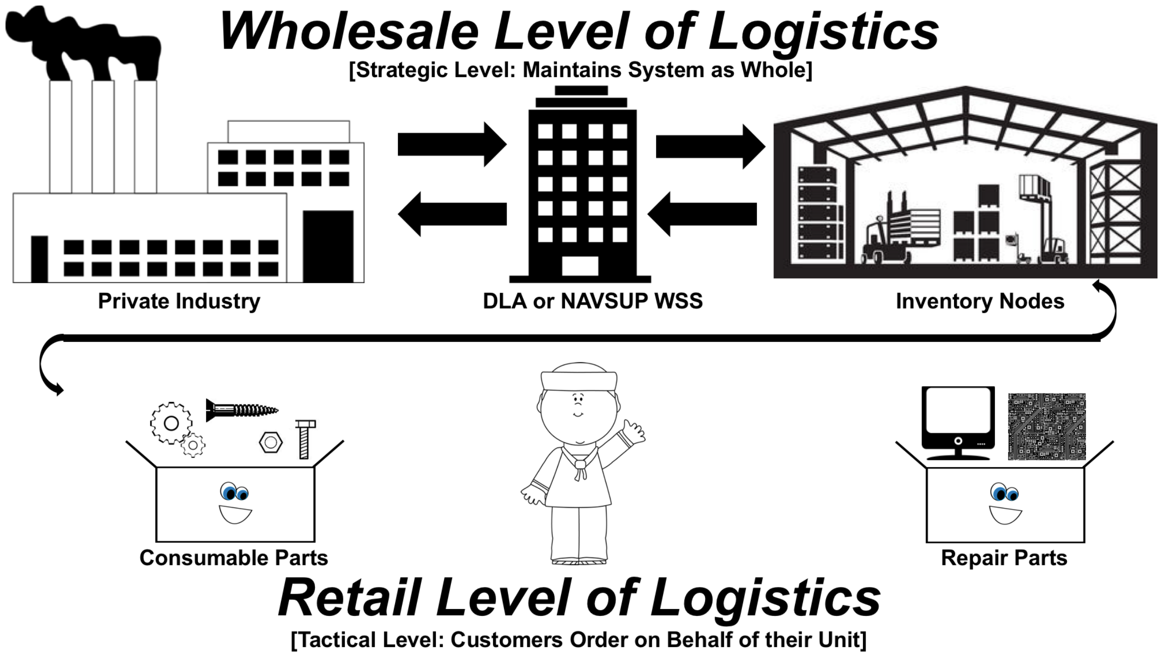


Figure was created from clip art in the public domain.

Figure 1. U.S. Navy Logistics Levels.

3. Supply System Example

We present the following example to illustrate the operation of the supply system. Suppose that the *USS Ronald Reagan* (CVN-76) needs to order consumable gasket materiel for a maintenance work center. A supply petty officer prepares a line of code called a MILSTRIP using the requisitioning software. This code contains the National Stock Number (NSN) of the requested item, quantity, and price (Naval Supply Systems Command [NAVSUP], 2015a). The first four digits of the NSN comprise the FSC code, which categorizes the item being ordered; in the present example it is packing and gasket materiel (NAVSUP, 2015a). The completed MILSTRIP is passed electronically to the Navy Enterprise Resource Planning (ERP) system, where the item is referenced to a cognizance symbol (COG) that determines whether the requisition is consumable or repairable. The requisition is then routed to DLA for consumable materiel or to NAVSUP WSS for repairable materiel (Naval Supply Systems Command [NAVSUP], 2015b). Because the materiel in the present example is consumable, the requisition is passed to DLA. DLA then refers the requisition to an inventory node for fulfillment.

Ideally, the requisition is filled within the timeframe mandated by NAVSUP Publication 485 and shipped to the customer via commercial transportation. Often, however, the requisition is not filled due to a wholesale system outage and the requisition is put on backordered status (NAVSUP, 2015a). The resulting delay may significantly impact the customer. Waiting for gasket materiel may leave an entire ventilation system inoperable, with tangible impacts on the crew and mission readiness. When materiel becomes available, the requisition is filled by the first available inventory node. *USS Ronald Reagan* then receives the gasket materiel and electronically acknowledges receipt, thus completing the requisition process.

B. OBJECTIVE

Because consumable parts directly impact overall fleet readiness, the Navy recognizes the need to identify consumable NSNs that cannot meet customer requirements in the mandated response time allotted. A particular NSN may be requisitioned without undue delay for a period of time only to suffer an outage for an extended time shortly after. When a NSN fails to meet customer requirements, it is colloquially referred to as a "Bad Actor."

The purpose of this thesis is expand the definition of a "Bad Actor" beyond colloquial terms and establish new metrics and rules to formally identify items of concern and classify them as either *NSNs at Risk, Bad Actors*, or *Bad Actors with Trend*. Essentially, we define these categories as follows:

1. *NSNs at Risk* are items that are not yet categorized as *Bad Actors* but are trending in a worsening direction throughout a particular year.

2. *Bad Actors* are those items that are failing to meet customer requirements in the mandated time required during a particular year.

3. *Bad Actors with Trend* are a subset of *Bad Actors* that also exhibit a worsening customer response time trend throughout a particular year. Of the three categories, *Bad Actors with Trend* contains the items of greatest concern, and should be emphasized the most in communications with DLA.

Our primary goal is to provide Naval Supply Systems Command (NAVSUP) with a classification of its consumable inventory that will be useful in its efforts to improve the wholesale inventory position through a dialogue with DLA.

C. THESIS ORGANIZATION

This thesis is the first known attempt to tackle the topic of identifying and classifying consumable NSNs of concern in the context of the military supply chain. In Chapter II, we examine two previous studies that reviewed DLA's general inventory management practices and offered their recommendations for improvement. In Chapter III, we explore in detail the data and methodology used in this thesis. We first introduce the thesis data set, discusses shortcomings in existing NSN analysis metrics, and define two new metrics, one for customer wait time, and the other for demand variability. We then characterize in detail the regression tree models and the non-parametric correlation test used to identify a statistical trend in the residuals, and fully define the three categories of troubled items introduced in Section B. In Chapter IV, we describe the results of three separate regression trees built on the basis of FSC code 5331 (O-Rings), FSC code 4930 (Lubrication and Fuel Dispensing Equipment), and FSC code 1285 (Fire Control Radar Equipment) and identify the items of concern that were found within each FSC code. In addition, we also analyze each unique FSC code with sufficient depth to determine the aggregate impact of items of concern to the U.S. Navy. In Chapter V, we conclude the thesis, offer recommendations for policy changes, and explore opportunities for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW

A. UNITED STATES GOVERNMENT ACCOUNTABILITY OFFICE REPORT

In 2014, the United States Government Accountability Office (GAO) analyzed DLA's inventory management practices, systems, and goals. All inventory systems, military or otherwise, must manage a critical balance between customer service, cost, and internal efficiency (see Figure 2). For DLA, excessive focus on cost or internal efficiency deprives the warfighter of required parts in a timely manner. On the other hand, excessive focus on customer service requires high inventory levels, which can compromise effectiveness in other areas that must compete for limited resources. GAO recommended that DLA develop metrics for service, cost, and internal efficiency and then manage its inventory system in a sustainable balance (GAO, 2014).

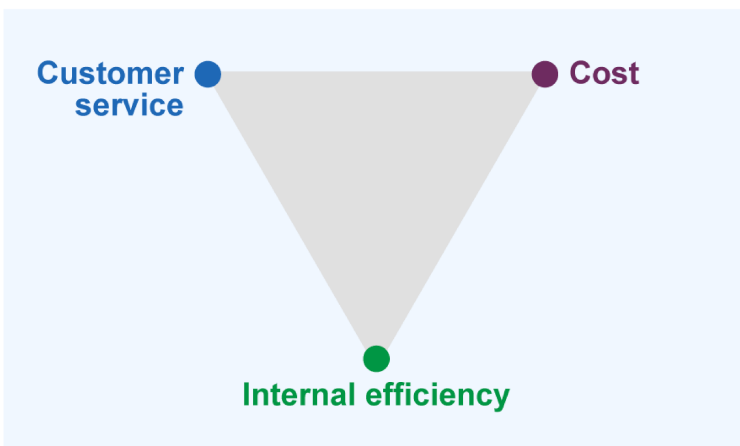


Figure 2. Competing Factors in DLA Inventory Management. Source: GAO (2014).

With regard to attaining a sustainable balance, GAO also examined DLA's progress in disposing of excess inventory. As recounted by GAO, DLA commissioned the private contractor LMI in 2008 to develop a mathematical model to identify potential excess materiel. Based on the model that it developed, LMI proposed setting inventory levels as a function of holding and repurchase costs. In 2009, DLA incorporated the LMI

model into its procurement practices and has continued to modify the model to reflect higher storage costs and other factors. DLA set a goal to dispose of $6 billion of excess inventory by FY2017 in order to reduce warehouse storage costs and to protect the value of its working capital fund (GAO, 2014).

GAO reported that DLA was making progress toward its inventory reduction goals. Specifically, when examining the combined Land, Maritime, and Aviation inventories as shown in Figure 3, total reduction of inventory for FY2012 to FY2013 was approximately $950 million for items with 1 to 4 years of no demand, roughly $460 million for items with 5 to 10 years of no demand, and nearly $200 million for items with 10 years or more of no demand (GAO, 2014).

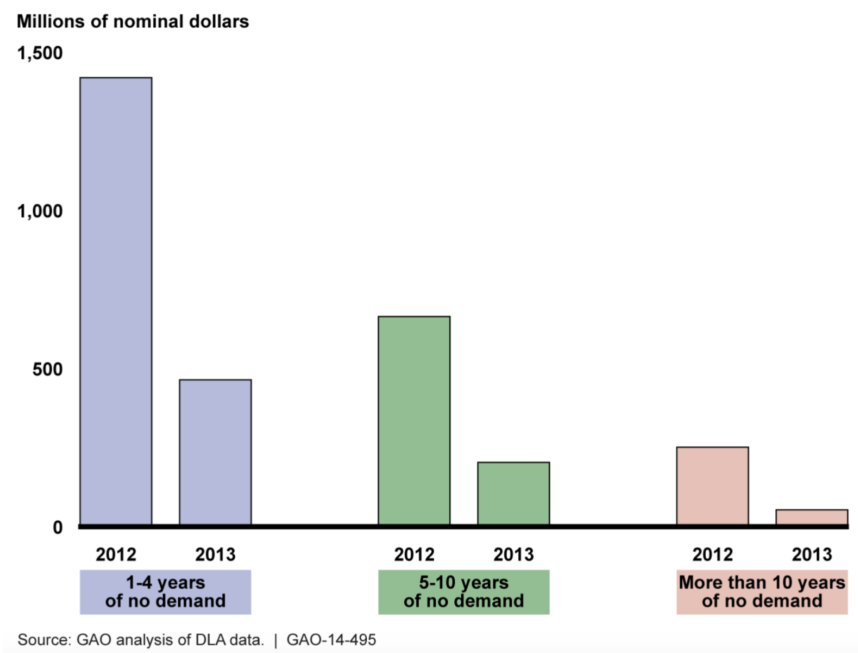


Figure 3. Value of DLA Land, Maritime, and Aviation Inventory with No Recorded Demand; FY2012 versus FY2013. Source: GAO (2014).

However, GAO cautioned that DLA may be disposing of materiel too aggressively in order to meet an arbitrary target value. It recommended that DLA continue to review its overall excess inventory goal and associated timeline, in order to minimize risk of inventory re-purchase at a higher cost in the future (GAO, 2014).

GAO also highlighted DLA's current inventory planning techniques. DLA has established criteria to place NSNs within one of the available techniques (see Table 1). Items with frequent, regular demand are subject to traditional demand forecasting techniques based on past demand history. Items with an irregular demand pattern pose a substantial challenge to any inventory system. In response, DLA in 2013 adopted two different statistical models to set inventory levels for low-demand and highly-variable demand items, respectively. Both models analyze the last five years of inventory data and set recommended minimum and maximum inventory levels as a function of backorder risk, cost of the item, and time between orders. Named "Peak" and "Next Gen," these two optimization calculations are held by a contracting vendor as proprietary, creating a challenge for DLA to critically assess its output and explore opportunities for model improvement.

A less commonly used but potentially powerful planning tool is collaborative forecasting, where a customer, such as a Navy shipyard or Navy TYCOM, partners with DLA staff to produce a more refined forecast based on past and expected future needs, using information usually unavailable to DLA under its other planning methods (GAO, 2014). Collaborative forecasting between NAVSUP and DLA already is available to improve the wholesale inventory position of the items of concern found in this thesis.

Table 1. Available DLA Inventory Planning Tools. Source: GAO (2014).

Approach	Key criteria for using approach for an item	Description	Approximate number of items[a]
Demand planning system	Demand in greater than 50 percent of quarters over the last 5 years, with a coefficient of variation between those quarters of less than or equal to 1.	Simple and advanced forecast modeling, based on historical demand, determines the amount of inventory needed over time.	200,000
Next-Gen	Demand in more than 50 percent of quarters over the last 5 years, with a coefficient of variation between those quarters greater than 1.	Optimization model evaluates several metrics to determine the minimum and maximum level of inventory for items.	137,000
Peak	Demand in less than 50 percent of quarters over the last 5 years.	Computation establishes a minimum and maximum level of inventory for the item.	334,000
Stock Keeping Unit (SKU) Build	Item must have at least two demands in at least 1 month over the last 12 months in order to be stocked at the location. These items do not meet the criteria to use the demand planning system and are not suited for Peak methodology due to multiple location management.	Computation establishes the amount of inventory stocked at each location over time based on the activity at that particular location.	834,000
Collaboration	Items with a collaboration agreement with the customer, such as demand data exchange.	Customer input, rather than historical demand, determines the amount needed over time.	38,000

Source: GAO analysis of Defense Logistics Agency (DLA) information.

[a]Item numbers are as of March 2014 and can fluctuate over time.

Approximate number of items field identifies the number of NSNs under each planning method.

B. RAND CORPORATION REPORT

The RAND Corporation (Peltz et al., 2015) also reviewed DLA's inventory management practices. While GAO focused heavily on reducing excess inventory, the RAND study focused on supply chain agility, or in other words, the ability to respond to highly variable customer demand. The DLA inventory is subject to highly irregular demand patterns, and despite efforts to develop better statistical forecasting models, the

authors argue that supply chain agility is the best solution to improve the responsiveness and efficiency of this highly dynamic inventory system.

To improve agility, Peltz et al. (2015) offer two core recommendations. The first is to reduce lead times in three aspects of the procurement process. First, the contracting process should minimize administrative lead time (ALT). Once a DLA procurement team writes a purchase request for a specific NSN to be obtained through a commercial supplier, the request must navigate through DLA's contracting section for solicitation and award. The requirements for federal government contracting are strictly prescribed by the Federal Acquisition Regulations (FAR), with additional regulations at the Department of Defense (DOD) and DLA levels. The RAND authors see opportunity to eliminate duplicate or cumbersome DOD and DLA regulations that add little value to the contracting process and create undue delay. In addition, the authors recommend the expansion of automated purchasing for frequently demanded items with little variation in purchase costs, which leverages an automated system to complete the contracting process with little human involvement, as a method to substantially minimize ALT (Peltz et al., 2015).

As a second lead time reduction strategy, the RAND study recommends that DLA incorporate production lead time (PLT) targets in contracts with commercial suppliers. This is a best practice from the private sector that rewards suppliers for fulfilling the requirement within a mutually-agreed production time. Peltz et al. (2015) noted that DLA only gauged expected future PLT by the PLT associated with the last contract on the item, which itself is a number completely self-generated by the private supplier during the contract process. Instead, if DLA incorporates PLT goals in written contract solicitations with financial rewards, suppliers will compete not only on the basis of cost but also time, and PLT will be reduced in the system. However, DLA should integrate PLT goals on a case-by-case basis, to ensure that the time savings benefit to the customer outweighs the added cost of the contract.

As a final lead time reduction strategy, the RAND study recommends expanding the use of long-term contracts (LTCs) of a type known as "indefinite delivery indefinite quantity" (IDIQ). A LTC establishes a requirement for a specific item for a given time

period, but does not specify a delivery quantity or schedule. Thus, the customer is free to request the item at will while the LTC is in effect, and the supplier is required to deliver the item in a reasonable time period. By reducing ALT in each purchase to near zero, which creates a positive opportunity cost savings in contracting manpower to instead focus on more complex contracting requirements, and by right-sizing order quantities, which avoids building excess inventory, the RAND authors found that LTCs are the most effective of the three lead time reduction strategies. In addition, the authors also found that NSNs with the most frequent demand patterns would benefit most from being on a LTC (Peltz et al., 2015).

The second core recommendation of the RAND study is to expand the information flow between the services and DLA, including the use of collaborative forecasting, which also is mentioned by GAO (2014). In addition, changes to weapons systems and their associated modified engineering and logistics requirements should be more effectively communicated by the services to DLA. Peltz et al. (2015) recommend establishing an information repository so that DLA managers can be aware of the potential risk of item obsolescence, and react accordingly in their procurement behavior.

Peltz et al. (2015) note that DLA had already been making progress in the direction of the two core recommendations prior to the RAND study.

III. DATA AND METHODOLOGY

A. INFORM-21 DATASET

The NAVSUP Inform-21 Database is the source of data for this thesis. Each Navy requisition is stored in this database along with amplifying information, and Inform-21 is continuously updated as requisitions are cancelled, shipped, or received. Inform-21 constitutes an archive of all requisitions after the Navy transitioned to ERP in 2010 (May, 2014).

This thesis is focused on consumable requisitions supporting readiness of Naval units, which limits the scope of data that we consider (see Table 2). Our objective is to capture a recent history of original consumable requisitions in the supply system, excluding such factors as local storeroom issues and subsequent follow up requisitions that would only serve to obscure the data. Applying the filters in Table 2 reduces the scope of the data set from over 11 million requisitions to roughly 3 million requisitions.

Table 2. Inform-21 Data Used to Support the Thesis Research

PARAMETERS	FILTERS
Requisition Time Period	January 1 2013-December 31 2015
COG	9B, 3B
FSC	FSC Codes < 6500
Storeroom Issues	Local Storeroom Issues Excluded
Follow Up Requisitions	Follow Up Requisitions Excluded
Cancelled Requisitions	Cancelled Requisitions Excluded
Pending Stows	Pending Stows Excluded
Acquisition Advice Codes (AAC)	Centrally Managed, Stocked, and Issued [AAC C&D] Stocked, but Future Procurement not Authorized [AAC V] Items which may be Required Intermittently [AAC Z]

AAC Codes are defined in NAVSUP (2015b).

B. COMPUTATIONAL AND STATISTICAL TOOLS USED

With a dataset originally exceeding 11 million requisitions, the most efficient option is to use scripting languages for data filtering, additional computation, and analysis. We create original scripts in Python and R to accomplish our objectives (to view the scripts in their entirety, see Appendix A and B). For the data filtering and additional computation, we execute our first script in the Python software environment to achieve the filtering here in Chapter III, Section A, and develop new metrics for customer wait time and demand variability (Enthought Inc., 2016). For the statistical analysis, we ran our second script in the R software environment (R Core Team, 2015) to build regression trees, conduct a non-parametric correlation test on the resulting residuals, and classify items of concern into the categories of *NSNs at Risk*, *Bad Actors*, and *Bad Actors with Trend*.

C. THE NEED FOR IMPROVED METRICS

1. Time as a Critical Metric

Earlier internal studies at NAVSUP attempted to identify *Bad Actor* NSNs based on the frequency of requisitions in backordered status. But from a customer's perspective, it does not matter if a requisition was backordered for a period of time as long as the requisition is filled within a timely manner. It therefore is reasonable to formulate performance metrics for the supply system using the time that it takes to fulfill requisitions. Although the Navy currently uses average customer wait time (ACWT) as a metric, it does not incorporate the priority level at which a requisition is made. We propose a new metric that takes into account both the time to fulfill a requisition and the time allowance for that requisition depending on the priority level and geographic location of the customer.

As prescribed in NAVSUP Publication 485, the Uniform Materiel Movement and Issue Priority System (UMMIPS) standards provide allowances for every stage of the requisition process, including the total order-to-receipt time for a requisition given its order priority and geographic location of the customer (NAVSUP, 2015a). As shown in Table 3, requisitions are divided into three priority bins—high (TP 1), medium (TP 2),

and low (TP 3). A mission-critical requirement almost certainly will be a high-priority order, while stock replenishment requirements almost certainly will be a low-priority order. As shown in Table 3, requisitions also are stratified into five geographical categories. Orders from within the continental United States (CONUS) are prescribed the tightest timetables, while those in hard-to-reach areas (geographic area "D," which corresponds to such locations as Diego Garcia and Djibouti) are allowed the most generous time-tables.

Table 3. UMIPPS Timetable. Adapted from NAVSUP (2015a).

UMMIPS TIME STANDARD IN CALENDAR DAYS																
	PD 01-03 ALL RDD's						PD 04-08, RDD 777 or PD 04-15 W/RDD 444, 555, 777					PD 04-15 W/Blank RDD or RDD >8 days after Reqn Date				
	TP 1 AREA						TP 2 AREA					TP 3 AREA				
PIPELINE SEGMENT	CONUS	A	B	C	D	EXP	CONUS	A	B	C	D	CONUS	A	B	C	D
A. Requisition Submission Time	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	1	1	1	1	1
B. ICP Processing Time	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	1	1	1	1	1
C. Storage Site (or Base) Processing, Packaging and Transportation Hold Time	1	1	1	1	1	1	1	1	1	1	1	3	3	3	3	3
D. Storage Site to CCP Transportation Time	N/A	1	1	1	1	N/A	N/A	3	3	3	3	N/A	7	7	7	7
E. CCP Processing Time	N/A	.5	.5	.5	1	N/A	N/A	1	1	1	1	N/A	5	5	5	10
F. CONUS In-Transit Time	1	1	1	1	1	N/A	4	2.5	2.5	2.5	2.5	9	7	7	7	7
G. POE Processing and Hold Time	N/A	1	1	1	2	N/A	N/A	2	2	2	3	N/A	5	5	5	10
H. In-Transit to Theater Time	N/A	1	1	1	1.5	3	N/A	1	1	1	1.5	N/A	5	12	19	27
I. POD Processing Time	N/A	.5	.5	.5	1	N/A	N/A	.5	.5	.5	1	N/A	3	3	3	5
J. In-Transit, Within-Theater Time	N/A	1	1	1	1	1	N/A	1	1	1	1	N/A	5	5	5	5
K. Receipt Take-Up Time	.5	.5	.5	.5	.5	.5	1	1	1	1	1	2	2	2	2	2
Total Order-to-Receipt Time	3.5	8.5	8.5	8.5	11	6.5	7	14	14	14	16	16	44	51	58	78

The bottom row of the table, Total Order-to-Receipt Time, prescribes the total allowed requisition times given geographic location and order priority.

By using the UMIPPS standard total order-to-receipt time (aka mandated customer wait time) highlighted in yellow in Table 3, each requisition in the data set is assigned a mandated order-to-receipt time. A new metric, which we call the customer time limit (CTL), is defined as follows:

$$Customer\ Time\ Limit\ (CTL) = \frac{Actual\ Customer\ Wait\ Time}{Mandated\ Customer\ Wait\ Time}\ x\ 100\%$$

Note that the CTL is unitless, which allows low and high priority requisitions ordered at different locations around the globe to be compared. If a requisition arrives earlier than its mandated time, the CTL is less than 100%; if a requisition arrives late, it is greater than 100%. Obviously, the customer desires a number less than or equal to 100%. This new metric has applicability beyond the scope of this thesis, and offers NAVSUP a more nuanced method to measure the responsiveness of the supply system.

CTL is calculated for each requisition using a Python computer language script. The histogram and summary statistics for CTL in CY2013 and CY2014 show a pronounced right tail (see Figure 4). The mean CTL during this period was 420%, but the mean is strongly affected by the long right tail of the distribution. The median CTL, less affected by the skewed distribution, was 142%—which still suggests that the inventory system is underperforming as a whole.

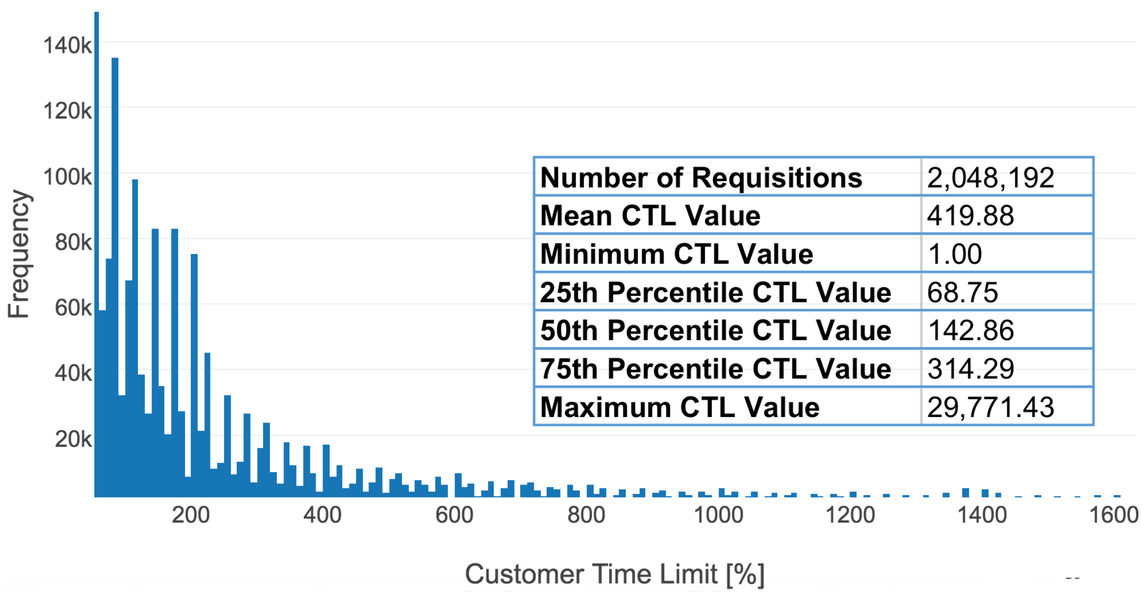


Figure 4. CTL Partial Histogram (less than 95th Percentile) and Summary Statistics, 2013–2014

To examine whether certain requisition characteristics correlate to CTL, we group the data by NSN. A median CTL value for each NSN is derived from its associated group of requisitions. Figure 5 gives a visualization of the median CTL of requisitions in the

form of a "heat map" in which each NSN is cross-classified by two criteria: the number of requisitions on the item in fiscal years 2013 and 2014, and the extended money value (quantity ordered times the unit price) over the same two years. The NSNs are then aggregated into grid squares and assigned a "heat color" according to the value of the median CTL in each square. The heat map clearly demonstrates that the worst performing NSNs tend to be the less frequently ordered, expensive items. It also shows that as order frequency decreases, an increasing number of grid locations in the respective column exhibit poor CTL performance.

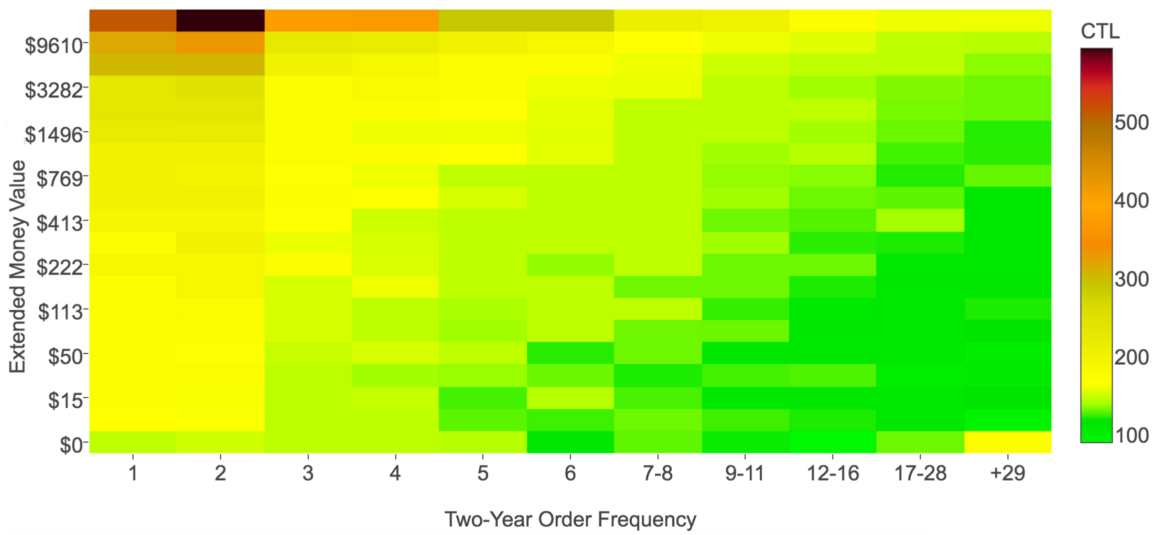


Figure 5. Heat Map, Median CTL by Grid Location, 2013–2014

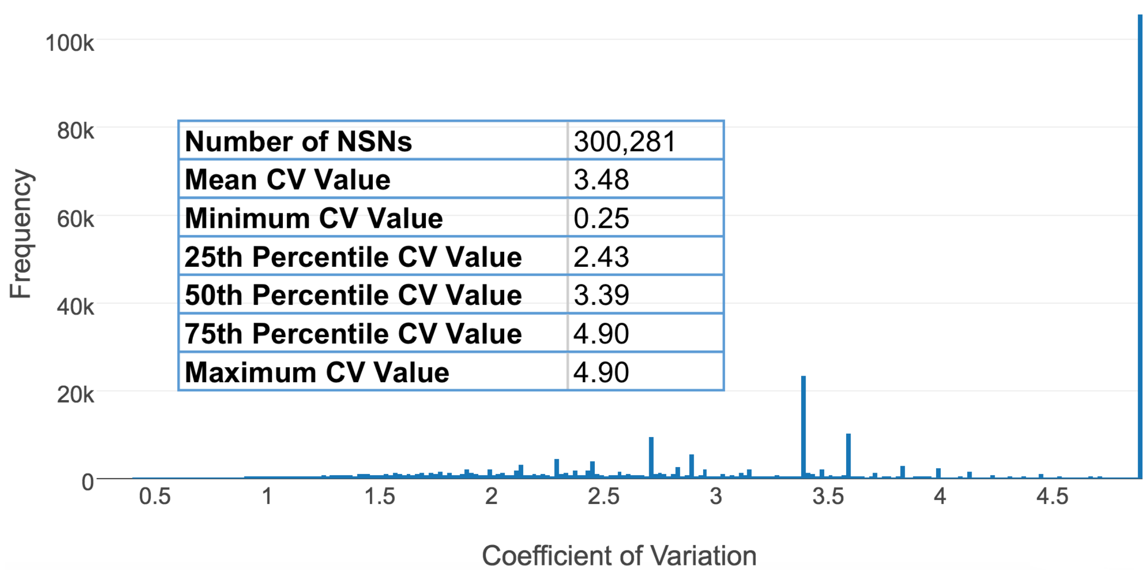
2. Coefficient of Variation as a Critical Metric

In Figure 5, the quantity of yellow and red grid locations correlate inversely with order frequency, implying that another critical metric is variability. The coefficient of variation (CV) is a statistical metric that is widely used in inventory management to measure variability. To calculate the CV, record the quantity demanded for an item on a particular basis (e.g. monthly or quarterly) over a certain time span (e.g. 1 year or 2 years). From the recorded values, calculate the sample standard deviation and sample mean, and the respective ratio of these two values constitutes the CV (Wackerly, Mendenhall, & Scheaffer, 2002). CV is also presented in equation form below.

$$Coefficient\ \ of\ \ Variation\ =\ \frac{Sample\ \ Standard\ \ Deviation}{Sample\ \ Mean}$$

Note that the CV is unitless and invariant under linear transformations of the variable in question.

Using a Python computer language script, the demand patterns of each NSN are recorded on a monthly basis over a two-year time span (CY2013 and CY2014), resulting in twenty-four observations that form the basis for a unique CV score per NSN. If an item is only ordered once in two years, by the nature of the CV calculation, the CV score will be 4.9 (sample standard deviation of roughly 0.2 divided by sample mean of approximately 0.04). 4.9 is the CV score assigned to roughly one-third of the NSNs in this dataset, far exceeding any other CV value in frequency (see Figure 6). Rigoni and Correia de Souza (2016) demonstrate that Navy-managed items may be forecasted as long as the CV score, as measured over a twenty-four-month period, is less than 2. As the same general forecasting techniques available to NAVSUP WSS are also available to DLA, we can apply their findings to this thesis. With the distribution of CV scores in Figure 6 heavily skewed to the right, DLA is managing an inventory system where more than 75% of NSNs cannot be forecasted.



Number of NSNs	300,281
Mean CV Value	3.48
Minimum CV Value	0.25
25th Percentile CV Value	2.43
50th Percentile CV Value	3.39
75th Percentile CV Value	4.90
Maximum CV Value	4.90

Figure 6. CV Histogram and Summary Statistics by NSN, 2013–2014

When the same technique used to produce the heat map shown in Figure 5 is applied in similar fashion to CV as the metric of interest, a heat map is produced for requisitions ordered in CY2013 and CY2014 (see Figure 7). Due to the nature of the CV calculation, less frequently ordered items are certain to have a higher CV score. However, the heat map does visually illustrate that the only grid locations that are "forecastable" (containing a median CV score of less than 2) are those with a minimum two-year order frequency of 12 (judging by the appropriate shade of yellow corresponding to CV < 2 first appearing in this column as one moves from left to right on the x-axis), which represent a minority of grid locations and once again emphasizes the challenge DLA faces in managing an inventory system with a high degree of demand variability.

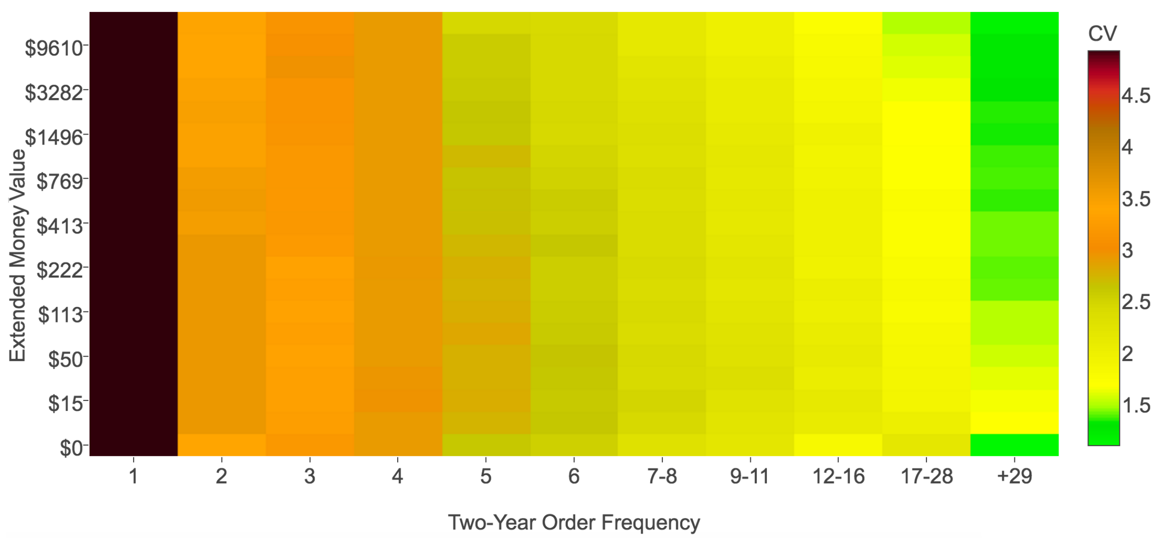


Figure 7. Heat Map, Median CV by Grid Location, 2013–2014

D. REGRESSION TREE METHOD

We have established Customer Time Limit (CTL) and Coefficient of Variation (CV) as critical metrics that should be incorporated into the identification and classification of items of concern. Next, we use statistical modeling to relate CTL to a set of predictor variables. We use the result of the statistical model to obtain residual values (differences between predicted values and actual values) as our primary parameter of interest, vice simply the predicted values, as is usual. We use the residuals as means to

examine factors exogenous to the model, indicating a trend that assists in classifying items for further scrutiny.

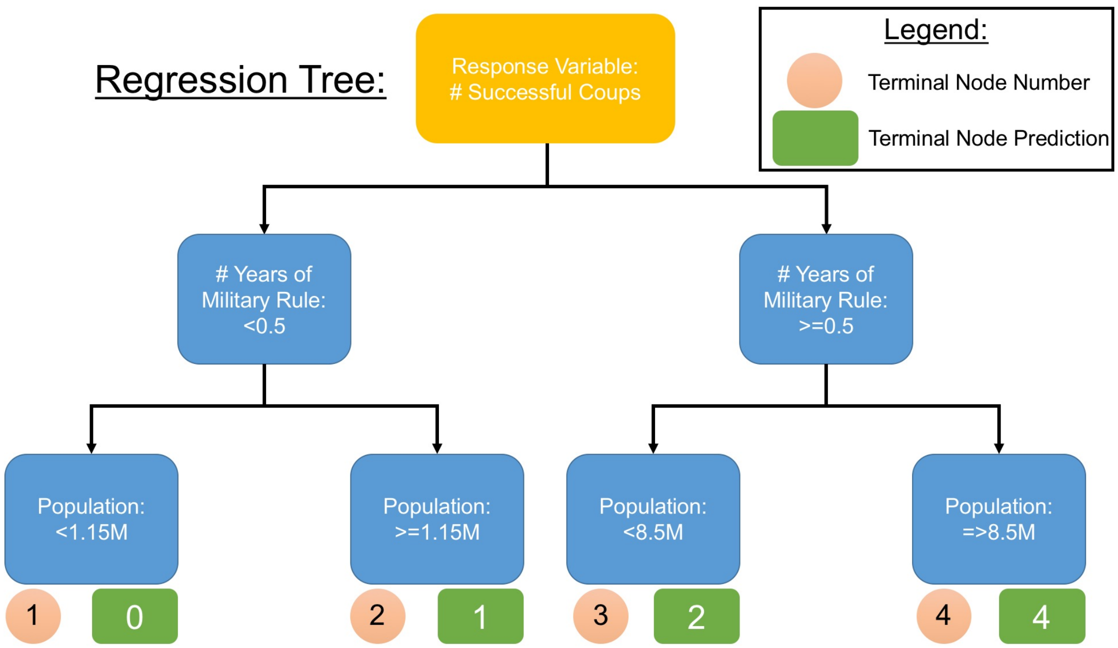
Classification and regression trees (CART), implemented in the R software environment via the RPART package (Therneau, Atkinson, & Ripley, 2015), provide a flexible approach to developing a nonparametric regression model. CART is able to identify interaction structures between the variables using a series of binary "splits" that vary depending on location in the tree structure. This allows CART to describe specialized relationships between variables in different parts of the data space (Breiman, Friedman, Olshen, & Stone, 1984).

At each step, the CART algorithm seeks to find a best split that achieves a separation of high and low values into respective nodes. As the tree grows, the cross-validated relative standard error of the model decreases, but after a certain point, there is little benefit from additional splitting due to sparsity of the data. An effective approach is to grow a complex tree, and then prune it to the point one step before the standard error is minimized, or in the minimal marginal benefit case, prune to the point where the slope of the standard error line begins to flatten (Breiman et al., 1984).

1. Regression Tree Example

To illustrate how CART is used, we present a simple example. In 1997, political scientists Bratton and Van De Walle published a study of post-independence African countries. In addition, they released their source data in the R statistical software environment, which described 47 Sub-Saharan African nations with nine numeric variables, including population, country size in thousands of square kilometers, years of post-independence military rule, and number of successful coups from independence to 1989. One intriguing application of CART is to set the the number of coups as the response variable, with the remaining eight variables as predictors.

Upon applying CART using the RPART package, four terminal nodes are found (see Figure 8). Despite having eight predictor variables available, RPART chooses two: the number of years of military rule, and population of the country. Each terminal node is an interaction between these two variables, and it is noteworthy that the population splitting rules are different for terminal nodes 1 and 2 and for terminal nodes 3 and 4. It would be difficult and cumbersome to discover this type of interaction structure using common linear regression techniques.



For simplicity, predicted values have been rounded to whole numbers.

Figure 8. Regression Tree Example: Large Tree

Figure 9 shows how the cross-validated relative standard error decreases as the complexity of the regression tree increases. It shows that the rate of decrease flattens considerably after the first split, with almost no marginal benefit from additional splits. The model is likely to be sufficient if pruned back to two terminal nodes.

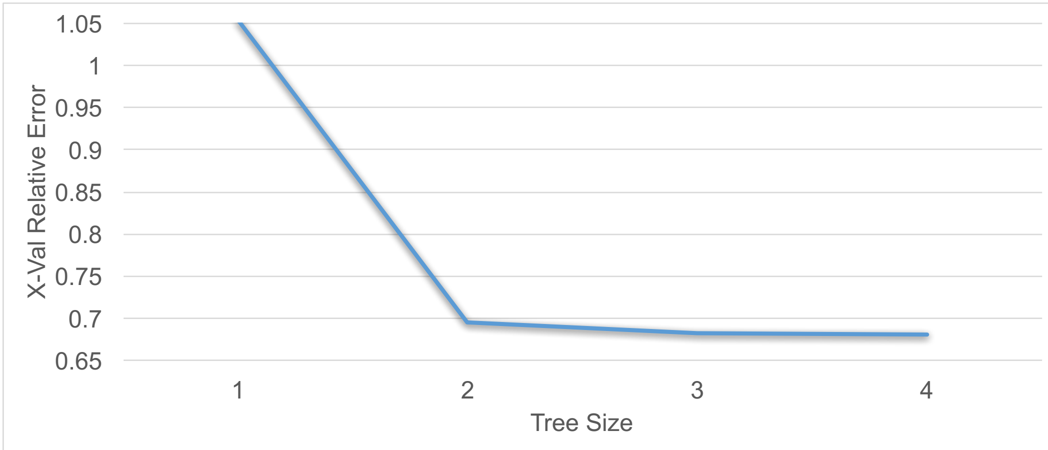
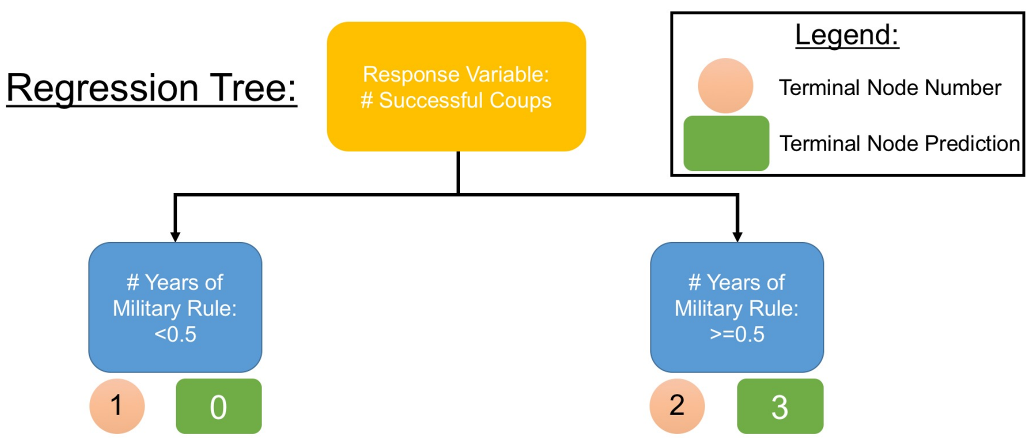


Figure 9. Regression Tree Example: Relative Standard Error as a Function of Tree Size

When pruned as shown in Figure 10, the tree shows that the most important predictor of number of successful coups is the duration of military rule, with a threshold for splitting set at 0.5 years (six months). The 23 countries that were subjected to military rule for less than six months had on average zero coups, and the 24 countries that were subjected to military rule for at least six months had on average 3 coups.



For simplicity, predicted values have been rounded to whole numbers.

Figure 10. Regression Tree Example: Pruned Tree

2. Regression Tree Method Applied to Inform-21 Dataset

We apply RPART in a similar manner to build large regression trees and then prune them back to an acceptable level of relative standard error (Therneau et al., 2015). The model will consider CY2013-2014 as the training set, and CY2015 as the test set. The training set is used to develop the model, and the test set is used to evaluate how well the model is able to predict new cases. The resulting residuals from applying the regression tree model to the test set is the basis for detecting the statistically significant trends over time in identifying and classifying items of concern.

E. REGRESSION TREE MODEL RESPONSE AND PREDICTOR VARIABLES

1. Customer Time Limit as Response Variable

Customer time limit (CTL) has been established as a critical metric, and is part of the basis in defining items of concern. Thus, CTL will be the variable that our regression tree models attempt to predict (aka the response variable). However, as previously shown in the histogram in Figure 4, this variable has a highly skewed right tail. Transforming a heavily skewed variable via a natural logarithm tends to create a more normally distributed variable, which in turn creates the potential for more statistically significant results (Wackerly et al., 2002). When the natural logarithm is applied to CTL a less skewed distribution is indeed obtained. In addition, the mean and median are now nearly identical (see Figure 11).

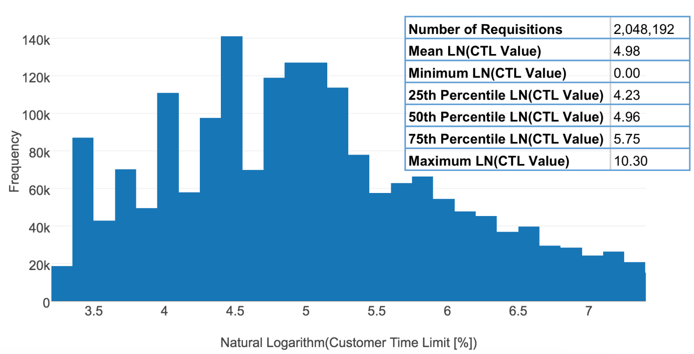


Figure 11. Partial Histogram for CTL (5th Percentile to 95th Percentile) Plotted on a Logarithmic Scale and Summary Statistics, 2013–2014

2. Predictor Variables

As shown in Table 4, Seventeen predictor variables are available to the regression tree. Only *UP* and *QUANTITY* are numeric, *DATE_ORDERED* is handled as a special date class, and the remaining variables are categorical. Some categorical variables require additional explanation. *PROJCODE* refers to a code that matches the requisition to certain special projects and cost information (NAVSUP, 2015b). *SUPPLYSOURCE* and *POE* (Point of Entry) are each three digit codes representing an inventory node somewhere in the world reflecting the node that filled the requisition and the node that first received the requisition, respectfully. *RDD* (Required Delivery Date) is set to 777 by default, but if the timetable for delivery is unsatisfactory, customers may enter a different code to indicate priority of shipment (NAVSUP, 2015b).

CV_CAT is a categorical variable that transforms the coefficient of variation into four categories: CV_ULTRALOW (<=1.0), CV_LOW (1.0< CV_LOW < 2.0), CV_HIGH (2.0< =CV_HIGH < 3.4), and CV_ULTRAHIGH (> 3.4). CV_ULTRALOW and CV_LOW represent forecastable items (those with a CV score of less than 2 per Rigoni and Correia de Souza, [2016]), while CV_HIGH and CV_ULTRAHIGH represent unforecastable items. The threshold between CV_ULTRALOW and CV_LOW is set at 1 (the middle value of the range), while the threshold between CV_HIGH and CV_ULTRAHIGH is set at 3.4 (the median CV value shown in Figure 6).

Table 4. Predictor Variables Available to Regression Tree Model

Predictor Variables	Description
NSN	National Stock Number of item ordered
IPG	Issue Priority Group: 1 [High Priority], 2 [Medium Priority], 3 [Low Priority]
Quantity	Quantity ordered in requisition
UP	Unit price of requisition
COG	Cognizance code of item: 9B, 3B
Date_Ordered	Date of requisition
Geozone_ordered	Two-digit geographic code corresponding to a region in the world where item was ordered
Geozone_shipped	Two-digit geographic code corresponding to a region in the world where item was shipped
Geozone_received	Two-digit geographic code corresponding to a region in the world where item was received
Priority	Similar to IPG, this notes the priority of the requisition from 1-15
BB	Records if a requisition entered backordered status: 1 [Yes], 0 [No]
PROJCODE	Project code of requisition
SUPPLYSOURCE	Three-digit code corresponding to inventory node where item was filled
POE	Three-digit code corresponding to inventory node where requisition was first received
SeriesCode	Single-digit code identifying CASREPS and other high priority orders
RDD	Required delivery date of requisition
CV_cat	Categorical Coefficient of Variation: CV_ULTRALOW, CV_LOW, CV_HIGH, CV_ULTRAHIGH

F. SEPARATE REGRESSION TREE MODELS FOR EACH FSC CODE

In our initial round of model building, we attempted to build a single regression tree model containing the roughly two million requisitions in the training set (which represents roughly two-thirds of total data), which resulted in explaining at best 30 percent of the variance in the training set data. It became apparent that some form of data division was required to give the CART algorithm an opportunity to build a more statistically significant model. Since DLA has primarily organized its materiel procurement by FSC code, our idea is to isolate the data for each FSC code and, on that basis, create a separate regression tree model per FSC code. Our objective is to give CART an ability to identify unique characteristics within the supply chain of each FSC code that would have been impossible when building a singular model for the entire data set. When a small number of FSC-specific regression trees are constructed, at least 70 percent of the variance in the training set data is explained, more than doubling the performance of the singular model.

As shown in Table 5, there are 310 distinct FSC codes represented in the data within CY2013 to CY2014. Discussing each unique separate regression tree models and their associate consumable items of concern in depth is beyond the scope of this thesis. Instead, we select three FSC codes that have an important impact on Naval combat readiness: FSC code 5331 (O-Rings; containing approximately 149,000 requisitions), FSC code 4930 (Lubrication and Fuel Dispensing Equipment; containing roughly 6,000 requisitions), and FSC code 1285 (Fire Control Radar Equipment; containing approximately 600 requisitions).

Table 5. Summary of FSC Codes Found in Inform-21 Dataset, 2013–2014

Item	Description
Number of Unique FSC Codes	310
Most Frequent FSC Code	5330 [Packing and Gasket Material]; 180K requisitions
FSC Code Near Median	5845 [Underwater Sound Equipment]; 553 requisitions
Least Frequent FSC Code	5630 [Nonmetallic Pipe and Conduit]; 1 requisition *
	* 10-way tie

G. SPEARMAN RANK CORRELATION COEFFICIENT AND TEST

The residuals of each regression tree model are used to identify NSNs with a worsening CTL trend over time. We use the Spearman rank correlation coefficient and its associated hypothesis test for this purpose. The Spearman rank correlation is the usual Pearson correlation coefficient but using ranks for two variables instead of the numeric value. An attractive property of the Spearman rank correlation coefficient is that it is invariant under increasing transformations of either or both of the variables. In our application this property is important because an increasing trend in the residuals over time need not be linear in the measured time scale. A treatment of the Spearman rank correlation and its use in testing the null hypothesis of no association between the two variables may be found in Myers & Well (2003). For our purposes, the two variables of interest are *DATE_ORDERED* (representing time), and the residuals (based on the CART models). The results are tested against the null hypothesis that time and residuals have no association. If the null hypothesis is rejected with a p-value of 0.05 or less, it represents a statistically significant positive trend and is part of the basis for classifying items of concern as either *NSNs at Risk* or *Bad Actors with Trend* (the third category, *Bad Actors,* does not consider statistical trends as part of its definition).

H. FORMALLY CLASSIFYING ITEMS OF CONCERN

In Chapter I, Section B, we introduced the three basic categories of items of concern: *NSNs at Risk*, *Bad Actors*, and *Bad Actors with Trend*. We now formally define each category. Each category is defined by three rules involving a particular range of 95% lower confidence bound of the median CTL value, and conducting or not conducting the Spearman test on the FSC code specific regression tree model residuals. In addition, after obtaining results from the first two rules, we apply the third rule and restrict the data to only those NSNs considered forecastable (CV score < 2; corresponding to variable values CV_ULTRALOW and CV_LOW). Each of these component will be fully defined in the subsequent sections. Our classification scheme method and associated generic rules, with particular settings for each category removed, is visually illustrated as Figure 12.

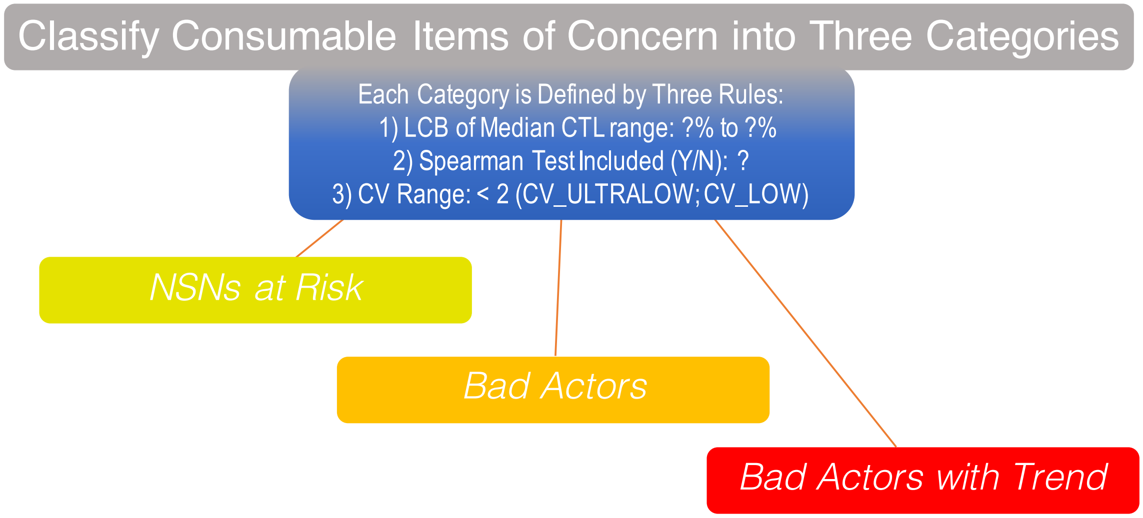


Figure 12. Consumable Items of Concern Classification Scheme

1. Customer Time Limit Used to Formally Classify Items of Concern

Customer time limit (CTL) is the foundation of all three categories, because it allows direct comparison between high and low priority requisitions ordered in different locations around the globe. But only using the predicted CTL values from the regression tree models to define items of concern is insufficient because the NSN is included as a predictor variable, which due to the structure of regression trees may have a particular NSN included in multiple terminal nodes. Thus, the predicted CTL value for a given NSN may return multiple results depending on the specific project code or backordered status contained in its associated requisition.

We also use CTL values from the test set to calculate 95% lower confidence bounds (LCBs) for the median CTL score per NSN, which are used to determine whether a NSN should be identified as an item of concern. We use a nonparametric 95% lower confidence bound (LCB) for the true population median consisting of the r^{th} largest sample value, where the value of r is determined from the binomial distribution based on the confidence level and sample size (Conover, 1999, p. 143). For example, if the sample size is $n = 200$ one obtains $r = 88$, and there is at least 95% confidence that the true population median is greater than or equal to the 88$^{\text{th}}$ largest sample value.

We now use our two statistical criteria (Spearman rank correlation test p-values and 95% LCBs for CTL) to classify each NSN according to our three categories of items of concern. *NSNs at Risk* represent items that are not yet *Bad Actors* but are statistically trending in that direction. Since they are not yet failing to meet customer requirements, we are interested in items with a minimum LCB of the median CTL value of 80 percent and a maximum LCB of the median CTL value of 99 percent. This range of LCB of the median CTL values, coupled with a statistical trend, should capture the NSNs on the verge of failing to meet customer requirements. *Bad Actors* and *Bad Actors with Trend* are already failing to meet customer requirements from a time-based standpoint. Thus, we assign a minimum LCB of the median CTL value of 100 percent, which demonstrates that these items are at best arriving at their maximum allowed customer wait times.

2. Spearman Test Results Used to Formally Classify Items of Concern

In Chapter I, Section B, we briefly mentioned the statistical trend required to classify items in the category of *NSNs at Risk* and *Bad Actors with Trend*, and here in Chapter III, Section G, we introduced the concept of the Spearman test. We desire to combine both concepts together to formally classify troubled items.

We apply the Spearmen test to each NSN in the subset of data applicable to the FSC code currently being modeled, testing each NSN for an association between time and its regression tree residuals, and recording a p-value for the significance of each NSN's result. For the *NSNs at Risk* and *Bad Actors with Trend*, a p-value of 0.05 or less indicates a statistically significant trend, so an upper bound of 0.05 in the Spearman test p-value is applied to assist in classifying these two categories. In the category of *Bad Actors*, we are indifferent to a statistical trend in the residuals, so this particular filter is not applied.

3. Coefficient of Variation Used to Formally Classify Items of Concern

As shown previously in Figures 5, 6, and 7, it is clear that DLA is managing an inventory system with a highly variable demand pattern. This variability would stress any commercial or military inventory system, so we recommend restricting the focus to only those items that are forecastable (a CV score of less than 2 corresponding to variable

values CV_ULTRALOW and CV_LOW). *CV_CAT* is included as a predictor variable in the regression tree models, so this additional restriction is not applied until after the model residuals are recorded. When the filter is applied to the items of concern, the scope of unique NSNs we consider is reduced by approximately 80%.

4. Application of the NSN Classification Rules

In order to apply this formal classification scheme, the test set data, which constitutes roughly 1 million requisitions from CY2015, first must be isolated by FSC code. After this step, a regression tree model specific to a particular FSC code is built. We use the result of the regression tree model to obtain residual values (differences between predicted values and actual values) as our primary parameter of interest, vice simply the predicted values, as is usual. We use the residuals as means to examine factors exogenous to the model, indicating a trend to be analyzed per the Spearman test.

Concurrently, we calculate the LCB of the median CTL value for each NSN in the test set by grouping actual CTL values together by NSN and applying the Conover (1999) method. The potential results are then filtered to only those items that are considered forecastable (a CV score of less than 2 corresponding to variable values CV_ULTRALOW and CV_LOW). Appropriate rules to each category of item of concern are applied and a record of *NSNs at Risk*, *Bad Actors,* and *Bad Actors with Trend* is created for that particular FSC Code (see Figure 13).

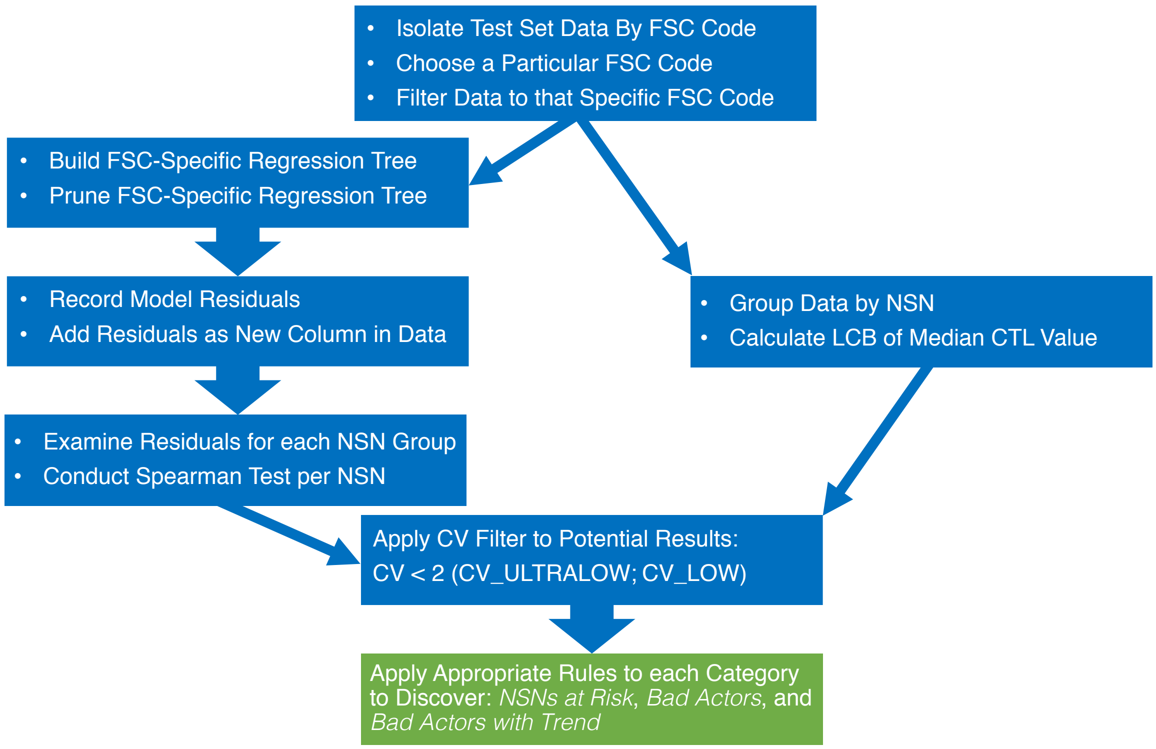


Figure 13. Formal Classification Scheme Process Flow Chart

The appropriate rules involving 95% LCBs of the median CTL and Spearman rank correlation test results, are particular to *NSNs at Risk*, *Bad Actors*, and *Bad Actors with Trend*. The rules are summarized in Table 6.

Table 6. Consumable Items of Concern: Categories and Associated Rules

Category	LCB of the Median CTL	Spearman Test Included
NSNs at Risk	80% to 99%	Yes
Bad Actors	at least 100%	No
Bad Actors with Trend	at least 100%	Yes

A CV score of less than 2 corresponding to variable values CV_ULTRALOW and CV_LOW applies to all three categories.

IV. RESULTS

As established in Chapter III, Section F, in order to obtain more statistically significant results, we isolate the data by FSC code. We select three FSC codes that have an important impact on Naval combat readiness: FSC code 5331 (O-Rings; containing approximately 149,000 requisitions), FSC code 4930 (Lubrication and Fuel Dispensing Equipment; containing roughly 6,000 requisitions), and FSC code 1285 (Fire Control Radar Equipment; containing approximately 600 requisitions). We build a unique regression tree for each of the selected FSC codes and identify their associated items of concern under the process flow from Figure 13 and the appropriate rules from Table 6.

The items of concern found are presented with minimal discussion. We recognize that some items contribute more directly to overall combat readiness than others, and that within the scope of this research, we cannot distinguish between the two. In addition, we also recognize that the inventory system is highly dynamic, and the items identified for further scrutiny might no longer raise concern at a future time. Finally, we conclude the chapter by extending the analysis to each unique FSC code in sufficient depth to comprehend the aggregate impact of items of concern to the U.S. Navy.

A. O-RINGS (FSC CODE 5331)

1. Description of O-Rings

O-rings, which support a wide variety of weapons system aboard Navy ships, submarines, and aircraft, were the third most frequently requisitioned FSC code in CY2013-2014. As shown in Table 7, the monetary value of O-rings may be small, but their impact on Naval readiness potentially is large.

Table 7. O-Ring (FSC Code 5531) Summary, 2013-2014

Characteristic	Value
# Requisitions	148,933
Unique NSNs	7,330
Median Unit Price	$0.48
Amount Purchased	$3,531,531.00

2. Heat Map for O-Rings

Applying the O-ring's subset of NSNs to a grid of two-year order frequency versus extended money value, another heat map of median customer time limit is generated (see Figure 14). As in Figure 5, the general conclusions are the same. The worst performing grid locations tend to be low-frequency, expensive items and as order frequency diminishes, a greater number of grid locations in the respective column have worsening median CTL scores. In addition, the lower right corner of the heat map contains a few empty cells, indicating that none of the NSNs met the criteria for being located there.

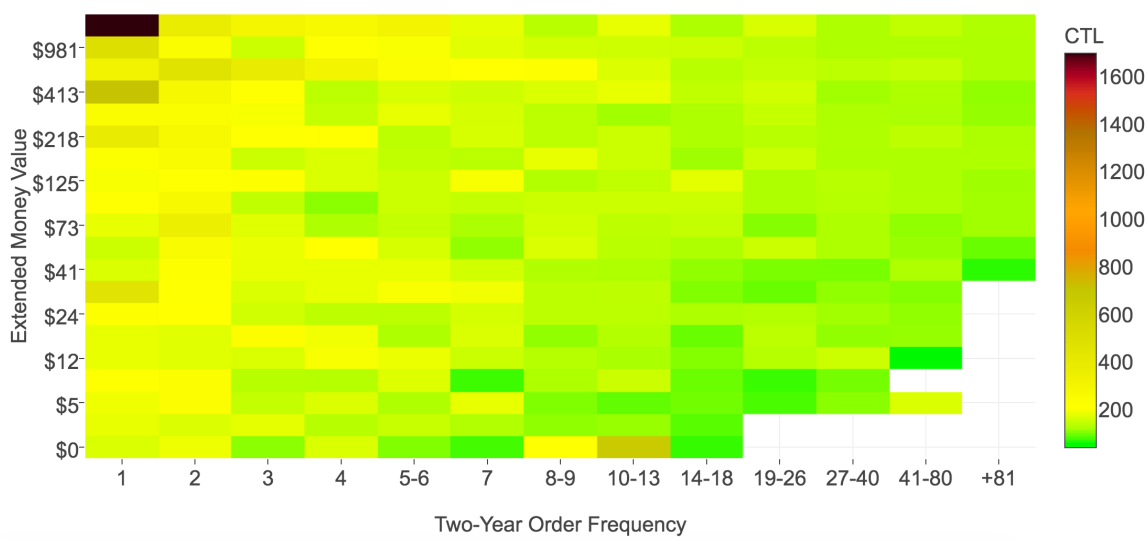


Figure 14. O-Rings Heat Map, Median CTL by Grid Location, 2013-2014

3. Regression Tree for O-Rings

After applying RPART to the natural logarithm of CTL, an initial tree is generated. The resulting relative standard error curve has an initial steep negative slope as the tree grows in size, but soon levels off to a nearly flat line (see Figure 15). The appropriate place to prune is somewhat arbitrary, but tree size fourteen appears to be a place where the marginal benefit from additional splits approaches zero.

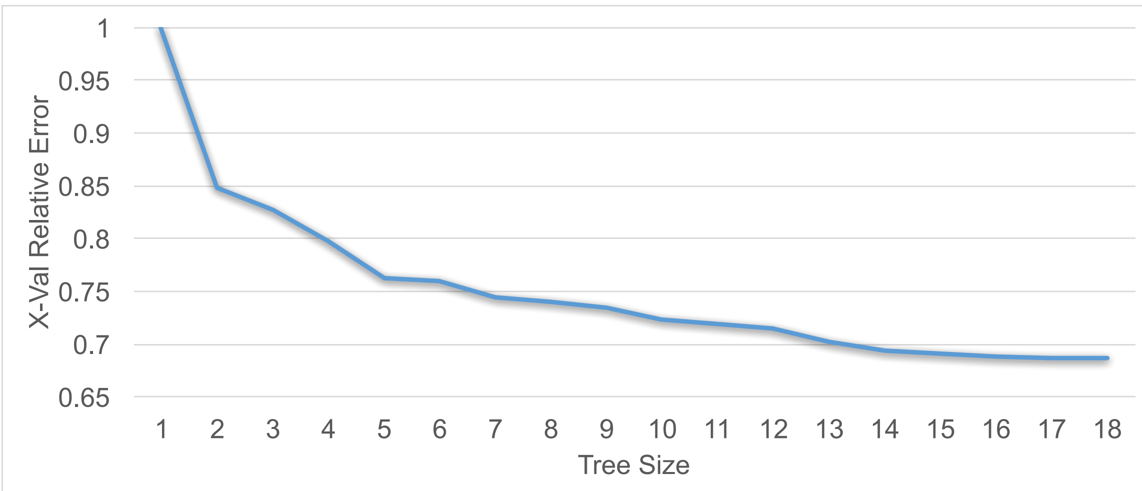
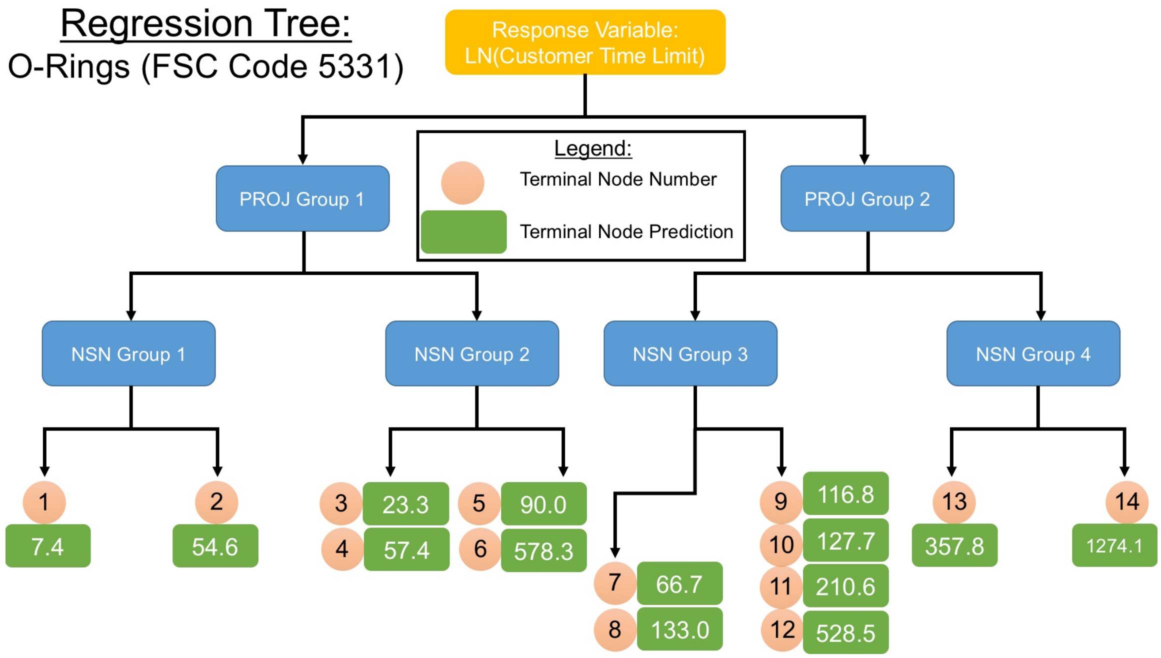


Figure 15. O-Rings Regression Tree: Relative Standard Error as a Function of Tree Size

Upon pruning the tree to fourteen nodes that involve six predictor variables the final model for O-Rings is created. The pruned tree is too extensive to be presented here, but it can be summarized by showing the first two layers and the terminal nodes below each branch and their associated prediction (see Figure 16).



Numbers in green boxes are geometric mean CTL values within the nodes.

Figure 16. O-Rings Pruned Regression Tree, Partial View

Terminal node 1 has the lowest predicted CTL of 7.4 percent when converted back to the original scale, and is an interaction between a group of approximately 200 NSNs and a single project code (705). We can only speculate why the NSNs in this terminal node are predicted with such rapid customer response times, but one possibility is that project code 705 is designated for materiel in a "Scheduled Repair/Overhaul Program" and may have benefitted from a collaborative demand forecast process between DLA and Navy shipyard representatives (NAVSUP, 2015b).

4. Items of Concern Belonging to the O-Ring FSC Code

a. NSNs at Risk Results

Using the residuals from the regression tree, and applying the appropriate filters from Table 6 to the data, a table of 18 *NSNs at Risk* is found (see Table 8). Within this FSC code, these results represent less than 1 percent in both NSN population and annual amount purchased.

Table 8. O-Ring *NSNs at Risk*, 2015

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL	CV
5331-00-576-9733	111	0.0013	114.3	85.7	CV_ULTRALOW
5331-01-033-2711	107	0.0396	85.7	81.3	CV_ULTRALOW
5331-00-579-7927	89	0.0394	100.0	85.7	CV_ULTRALOW
5331-00-480-2255	84	0.0227	114.3	87.5	CV_ULTRALOW
5331-00-103-1750	81	0.0440	93.8	85.7	CV_LOW
5331-00-338-1441	71	0.0223	156.3	97.7	CV_LOW
5331-01-461-1631	39	0.0032	171.4	93.8	CV_LOW
5331-00-936-6116	39	0.0000	85.7	85.7	CV_LOW
5331-00-115-1356	35	0.0098	114.3	85.7	CV_ULTRALOW
5331-01-005-0523	25	0.0459	112.5	81.3	CV_LOW
5331-00-252-6045	25	0.0002	142.9	87.5	CV_LOW
5331-01-009-7215	20	0.0003	125.0	93.8	CV_LOW
5331-01-289-9123	19	0.0026	142.9	85.7	CV_LOW
5331-01-189-3822	15	0.0427	114.3	85.7	CV_LOW
5331-01-330-9612	15	0.0025	171.4	85.7	CV_LOW
5331-00-248-3840	13	0.0450	142.9	85.7	CV_LOW
5331-01-277-7216	9	0.0048	942.9	81.3	CV_LOW
5331-01-024-9763	7	0.0181	107.7	81.3	CV_LOW

b. Bad Actor Results

We apply the appropriate filters from Table 6 to discover 157 O-ring *Bad Actors* (see Appendix C for complete table). Within this FSC code, this group represents roughly 2 percent of the total NSN population and about 10 percent of the total annual amount purchased. The table is too extensive to be viewed here in its entirety, but additional filtering within the CV category to CV_ULTRALOW alone reveals a list of 20 items, which are shown in Table 9.

Table 9. O-Ring *Bad Actors*, Partial Table (CV_ULTRALOW only), 2015

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL
5331-00-167-5122	346	1.000	214.3	214.3
5331-00-165-1962	144	0.003	156.3	114.3
5331-01-127-0971	88	0.003	125.0	100.0
5331-00-248-3837	87	0.999	112.5	100.0
5331-00-165-1970	80	0.741	178.6	106.3
5331-01-089-1583	65	0.998	200.0	171.4
5331-00-167-5141	59	0.702	128.6	114.3
5331-00-482-1595	50	0.645	171.4	152.9
5331-00-807-8993	46	0.880	196.9	142.9
5331-00-166-1020	44	0.815	182.9	100.0
5331-00-480-4733	39	0.669	290.9	156.3
5331-01-094-5959	29	0.477	121.4	100.0
5331-01-468-4214	28	0.334	171.4	136.2
5331-00-579-7543	28	0.936	153.6	114.3
5331-01-460-9039	26	0.965	247.3	193.8
5331-00-817-7783	17	0.939	173.1	118.8
5331-01-113-2084	12	0.700	209.4	118.8
5331-00-285-9842	11	0.442	228.6	106.3
5331-01-137-6897	7	0.560	218.8	145.5
5331-01-034-3464	5	0.729	150.0	100.0

c. *Bad Actors with Trend Results*

Classifying the subset of NSNs with a statistically significant trend from the list of 157 O-ring *Bad Actors*, 15 *Bad Actors with Trend* remain (see Table 10). Within this FSC code, these results represent less than 1 percent in both NSN population and annual amount purchased.

Table 10. O-Ring *Bad Actors with Trend*, 2015

NSN	Requisition Count	Spearman.P.value	CTL Median	LCB of Median CTL	CV
5331-00-165-1962	144	0.003	156.3	114.3	CV_ULTRALOW
5331-01-127-0971	88	0.003	125.0	100.0	CV_ULTRALOW
5331-01-181-2509	67	0.035	150.0	100.0	CV_LOW
5331-01-587-8959	65	0.000	128.6	100.0	CV_LOW
5331-01-007-1600	39	0.000	142.9	114.3	CV_LOW
5331-01-468-4209	25	0.001	185.7	136.2	CV_LOW
5331-00-689-6480	17	0.033	285.7	200.0	CV_LOW
5331-01-005-2305	16	0.044	214.3	153.8	CV_LOW
5331-01-093-3503	16	0.004	220.5	156.3	CV_LOW
5331-01-231-5217	12	0.018	192.9	171.4	CV_LOW
5331-00-763-2637	11	0.013	742.9	118.8	CV_LOW
5331-00-061-2209	9	0.005	781.3	181.3	CV_LOW
5331-01-206-6122	8	0.012	157.1	105.9	CV_LOW
5331-01-250-6735	8	0.000	107.1	107.1	CV_LOW
5331-01-399-8395	5	0.026	171.4	114.3	CV_LOW

d. Visual Example of a Bad Actor with Trend

The first two entries in Table 10 each contained more than 80 requisitions throughout 2015, making their scatterplots visually crowded and their trend difficult to discern. Instead we select a different entry from Table 10, an O-ring (NSN 5331-01-231-5217). In Figure 17, we can clearly see an association between time and residuals.

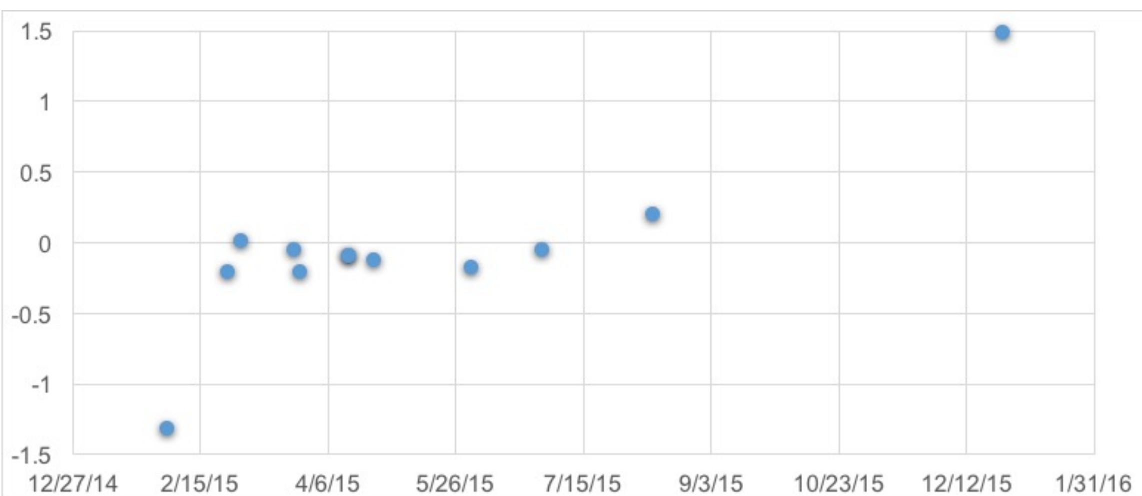


Figure 17. O-Ring (NSN 5331-01-231-5217) Scatterplot of Residuals, 2015

B. LUBRICATION AND FUEL DISPENSING EQUIPMENT (FSC CODE 4930)

1. Description of Lubrication and Fuel Dispensing Equipment

Lubrication and fuel dispensing equipment includes such items as handheld grease guns and fuel oil pumps. At first glance, these might seem like mundane parts, but grease guns in particular play a critical role in preventive maintenance. Without functioning grease guns, numerous high-dollar ship and aircraft systems are more prone to failure, thus directly negatively impacting mission readiness and the repair part budget. Lubrication and fuel dispensing equipment is slightly above the 75 percentile in most frequently ordered items per FSC code. In addition, with a median unit price of $70, and a bi-annual purchase amount of $3.7 million, the items in this FSC code exceeded the money spent on O-rings during the same time period despite containing far fewer unique NSNs (see Table 11).

Table 11. Lubrication and Fuel Dispensing Equipment (FSC Code 4930) Summary, 2013-2014

Characteristic	Value
# Requisitions	6,101
Unique NSNs	390
Median Unit Price	$70.36
Amount Purchased	$3,685,908.72

2. Heat Map for Lubrication and Fuel Dispensing Equipment

The subset of NSNs from the lubrication and fuel dispensing equipment code are grouped together, arranged as a grid of two-year order frequency and extended money value, and the median CTLs in each grid location are plotted as a heat map (see Figure 18). The conclusions from Figure 18 are no different than the other heat maps in Figure 5 and Figure 14. The worst performing grid locations tend to be the expensive, infrequently ordered items and as order frequency decreases, an increasing number of grid locations perform poorly. The vast majority of grid locations have a CTL value of 200 or higher, suggesting that this FSC code is underperforming as a whole.

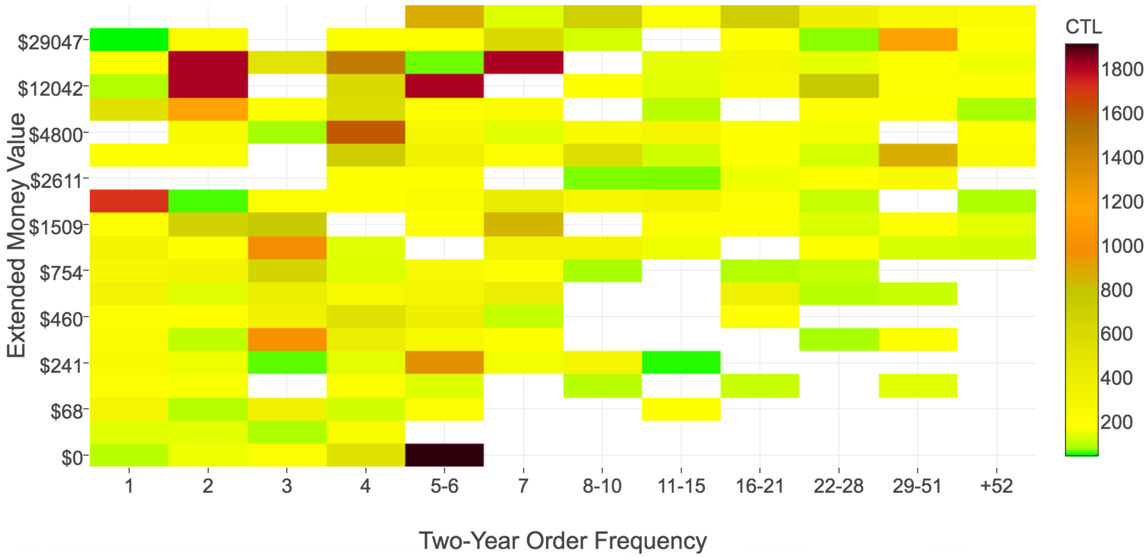


Figure 18. Heat Map, Lubrication and Fuel Dispensing Equipment Median CTL by Grid Location, 2013-2014

3. Regression Tree for Lubrication and Fuel Dispensing Equipment

After applying RPART to the natural log of CTL, an initial tree for lubrication and fuel dispensing equipment is created. The resulting relative standard error curve has an initial steep negative slope as the tree grows in size, but quickly levels off to a nearly flat line (see Figure 19). Tree size fifteen is the place where the relative standard error is minimized, but looking at the graph, there is little marginal benefit in a tree size greater than 4. In the interest of simplicity, we prune the tree to a size of four.

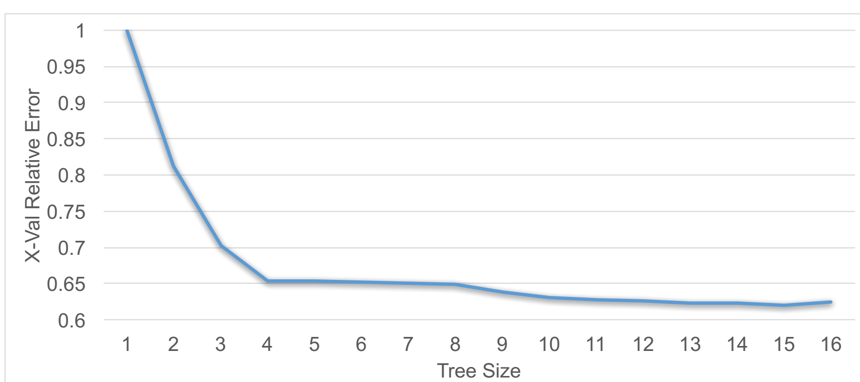
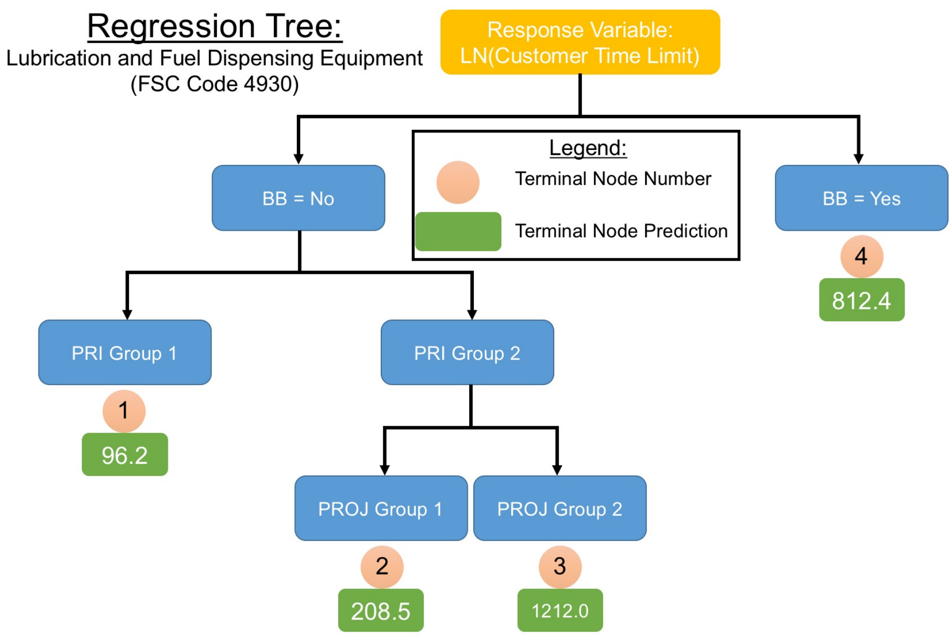


Figure 19. Lubrication and Fuel Dispensing Equipment Regression Tree: Relative Standard Error as a Function of Tree Size

Upon pruning the tree to four nodes that involve three predictor variables (BB, PRI, and PROJ) the final model for lubrication and fuel dispensing equipment is created. The pruned tree is presented in full as Figure 20. The tree structure makes intuitive sense since items that are backordered (BB=Yes) are generally predicted with larger CTL values than items that were never backordered. The one exception is terminal node three. The root cause is unknown to us, but one possible explanation is that almost half the project codes in this branch (ZH9, ZK6, ZQ0, and ZS0) are associated with initial outfitting of new weapons systems, which are commonly known to experience growing pains in their supply chain (NAVSUP, 2015b).



Numbers in green boxes are geometric mean CTL values within the nodes.

Figure 20. Lubrication and Fuel Dispensing Equipment Pruned Regression Tree, Complete View

4. Items of Concern Belonging to the Lubrication and Fuel Dispensing Equipment FSC Code

a. NSNs at Risk Results

When the *NSNs at Risk* criteria from Table 6 are applied, only one item is returned. NSN 4930-01-288-0866 (Nomenclature: Airline Lubricator) is shown in Table 12. Within this FSC code, this result represents less than 1 percent in both NSN population and annual amount purchased.

Table 12. Lubrication and Fuel Dispensing Equipment *NSNs at Risk*, 2015

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL	CV
4930-01-288-0866	19	0.008	100.0	88.6	CV_LOW

b. Bad Actor NSNs Results

We apply the appropriate filters from Table 6 to discover 16 lubrication and fuel dispensing equipment *Bad Actors* (see Table 13). Within this FSC code, these *Bad Actors* represent approximately 4 percent of all NSNs and roughly 25 percent of the amount annually purchased.

Table 13. Lubrication and Fuel Dispensing Equipment *Bad Actors*, 2015

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL	CV
4930-00-253-2478	112	0.6890	140.2	114.3	CV_ULTRALOW
4930-00-262-8868	74	0.8047	240.2	157.1	CV_ULTRALOW
4930-00-274-5713	57	0.2868	137.5	106.3	CV_ULTRALOW
4930-01-223-3730	35	0.9808	135.7	100.0	CV_ULTRALOW
4930-00-990-3330	27	0.7332	168.8	100.0	CV_ULTRALOW
4930-01-429-9930	18	0.7874	139.3	128.6	CV_LOW
4930-01-204-0634	17	0.0111	142.9	105.9	CV_LOW
4930-01-385-9025	16	0.1825	217.0	100.0	CV_LOW
4930-01-441-1313	15	0.8433	171.4	137.5	CV_LOW
4930-01-152-7902	14	0.9493	214.3	105.9	CV_LOW
4930-01-385-8946	11	0.9665	271.4	162.5	CV_LOW
4930-01-573-9597	9	0.4492	335.7	112.5	CV_ULTRALOW
4930-01-572-5645	8	0.9896	707.1	600.0	CV_LOW
4930-00-106-8674	6	0.8356	140.2	112.5	CV_LOW
4930-01-514-7828	6	0.8527	125.6	114.3	CV_LOW
4930-01-204-0638	5	0.0443	435.3	117.6	CV_LOW

c. *Bad Actors with Trend Results*

Out of the items in Table 13, only two exhibit a statistical trend. A Hose Strap Assembly (NSN 4930-01-204-0634) and a Hose Reel Strap (NSN 4930-01-204-0638) comprise the lubrication and fuel dispensing equipment *Bad Actors with Trend* and are shown in Table 14. Within this FSC code, this result represents less than 1 percent in both NSN population and annual amount purchased.

Table 14. Lubrication and Fuel Dispensing Equipment *Bad Actors with Trend*, 2015

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL	CV
4930-01-204-0634	17	0.011	142.9	105.9	CV_LOW
4930-01-204-0638	5	0.044	435.3	117.6	CV_LOW

d. *Visual Example of Bad Actor with Trend*

Both items in Table 14 have a statistically significant association between time and residuals. The Hose Strap Assembly (NSN 4930-01-204-0634) illustrates this point especially well, as the residuals are clearly trending higher by the end of 2015 (see Figure 21).

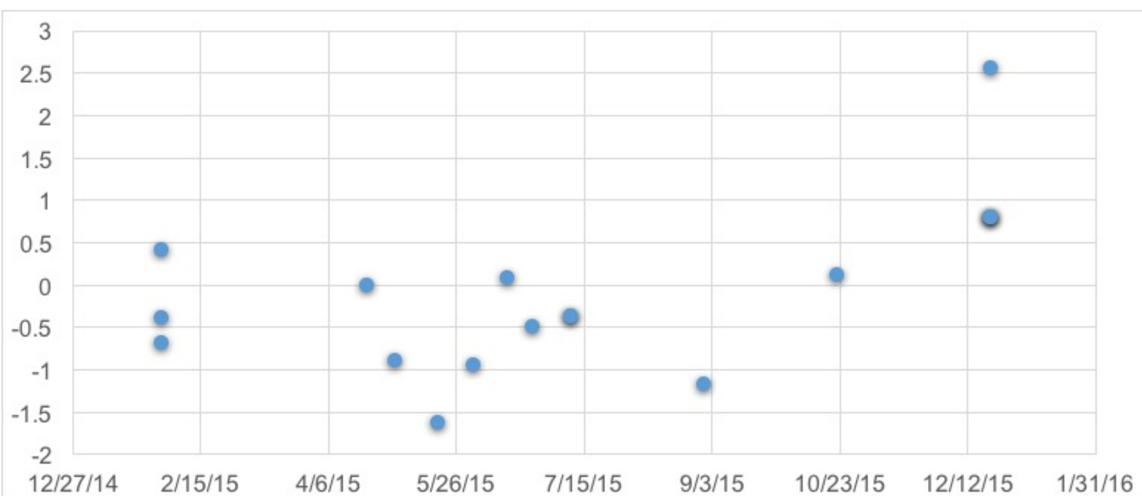


Figure 21. Hose Strap Assembly (NSN 4930-01-204-0634) Scatterplot of Residuals, 2015

C. FIRE CONTROL RADAR EQUIPMENT (FSC CODE 1285)

1. Description of Fire Control Radar Equipment

Fire control radar equipment is vital to safe navigation and enemy detection. The number of requisitions within fire control radar equipment is slightly above the median number of requisitions per FSC code. With a median price of $182, the items contained within this code are more expensive than typical consumables (see Table 15).

Table 15. Fire Control Radar Equipment (FSC Code 1285) Summary, 2013-2014)

Characteristic	Value
# Requisitions	620
Unique NSNs	52
Median Unit Price	$182.80
Amount Purchased	$510,635.90

2. Heat Map for Fire Control Radar Equipment

We apply the same technique for Figures 5, 14, and 18 to generate another heat map of median CTL value by grid location (see Figure 22). The relative scarcity of requisitions within this code ensures that most grid locations are empty. Of the grid locations exhibiting a heat color, there is no clear trend as order frequency decreases. The only apparent conclusion is that most of the grid locations have a median CTL value of 200 percent or higher, suggesting that this FSC code is underperforming as a whole.

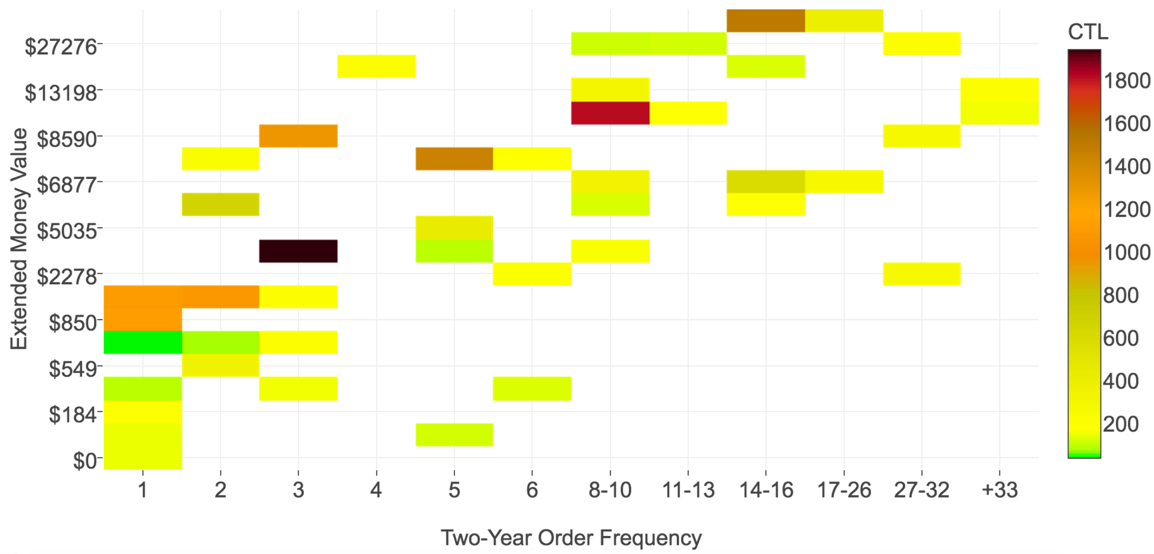


Figure 22. Heat Map, Fire Control Radar Equipment Median CTL by Grid Location, 2013-2014

3. Regression Tree for Fire Control Radar Equipment

After applying RPART to the natural logarithm of CTL, an initial tree is generated. The resulting relative standard error curve has an initial steep negative slope as the tree grows in size, but quickly levels off to a nearly flat line (see Figure 23). For the appropriate tree size, we desire to balance simplicity and an acceptable level of relative standard error. The marginal benefit in reduced error for tree size sixteen versus tree size ten is minimal, so we select tree size ten.

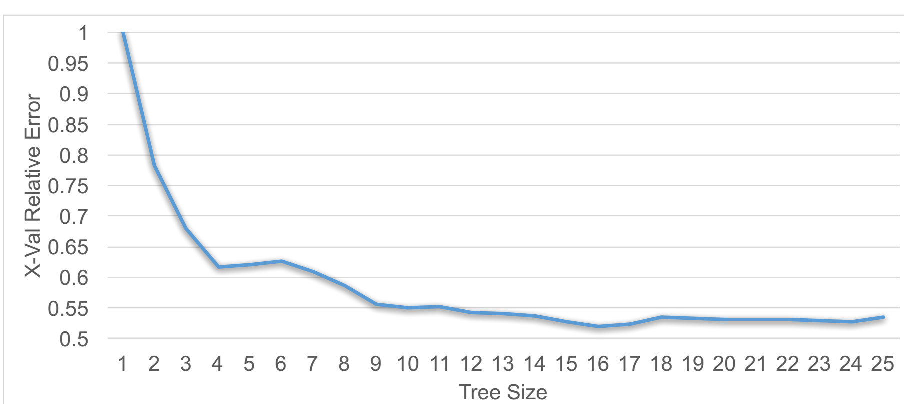
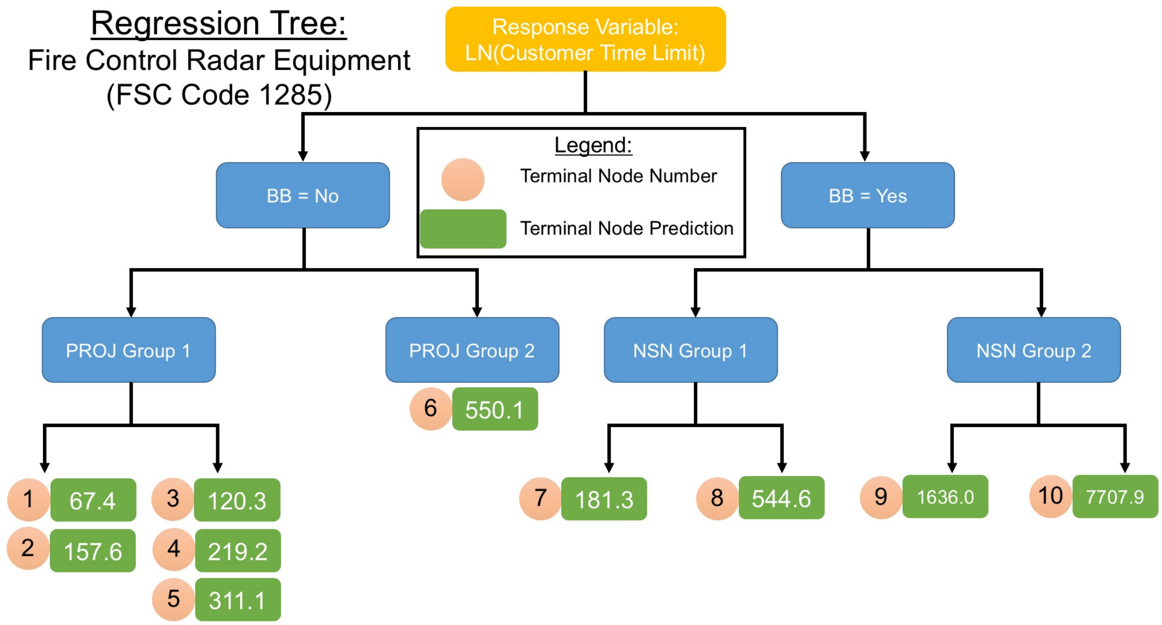


Figure 23. Fire Control Radar Equipment Regression Tree: Relative Standard Error as a Function of Tree Size

Upon pruning the tree to ten nodes that involve five predictor variables the final model for fire control radar equipment is created. The pruned tree is too extensive to be presented here, but it can be summarized by showing the first two layers and the terminal nodes below each branch and their associated prediction (see Figure 24).



Numbers in green boxes are geometric mean CTL values within the nodes.

Figure 24. Fire Control Radar Equipment Pruned Regression Tree, Partial View

4. Items of Concern Belonging to the Fire Control Radar Equipment FSC Code

a. NSNs at Risk Results

We apply the appropriate Table 6 criteria and obtain zero results.

b. Bad Actor Results

We apply the appropriate filters from Table 6 to discover 3 fire control radar equipment *Bad Actors* (see Table 16). Within this FSC code, these items represent roughly 6 percent of the total NSN population, and approximately 30 percent of the total annual amount purchased.

Table 16. Fire Control Radar Equipment *Bad Actors*, 2015

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL
1285-01-497-4884	17	0.286	314.3	271.4
1285-01-261-5539	14	0.000	129.3	128.6
1285-01-491-4985	14	0.896	164.3	127.3

CV score for all items is between 1 and 2, corresponding to the variable CV_LOW.

c. *Bad Actors with Trend Results*

Out of the items in Table 16, only one exhibits a statistical trend. An Electrical Grounding Hook (NSN 1285-01-261-5539) constitutes the only fire control radar equipment *Bad Actor with Trend* and is shown in Table 17. Within this FSC code, this result represents roughly 2 percent of the NSN population and roughly 1 percent of the annual amount purchased.

Table 17. Fire Control Radar Equipment *Bad Actors with Trend*, 2015

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL
1285-01-261-5539	14	0.000	129.3	128.6

CV score for item is between 1 and 2, corresponding to the variable CV_LOW.

d. *Visual Example of Bad Actor with Trend*

The association between time and residuals for the Electrical Grounding Hook (NSN 1285-01-261-5539) from Table 17 is illustrated in Figure 23 with a clear trend as the year progresses (see Figure 25).

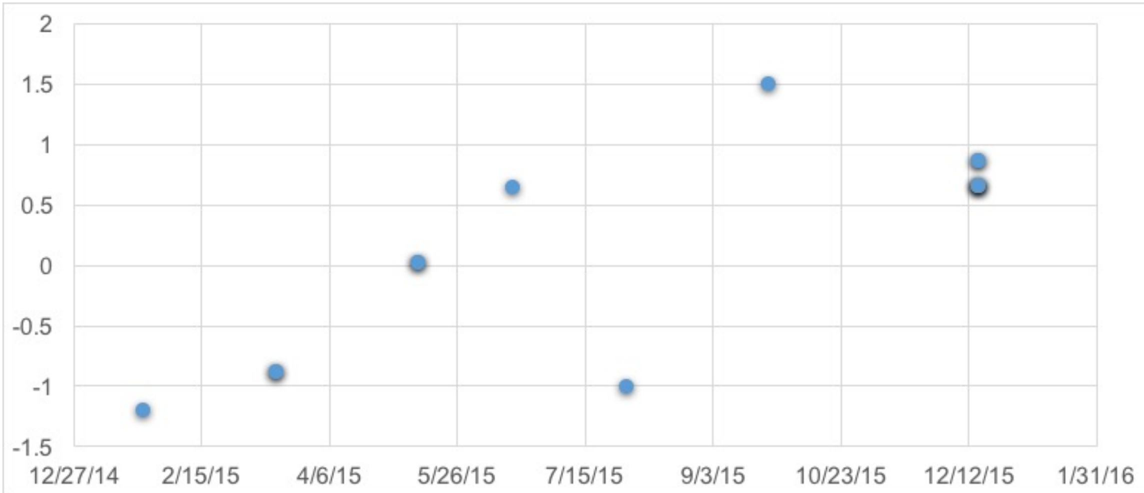


Figure 25. Electrical Grounding Hook (NSN 1285-01-261-5539)
Scatterplot of Residuals, 2015

D. QUANTIFYING IMPACT FROM ITEMS OF CONCERN

We have examined NSNs from within three different FSC codes vital to Naval combat readiness and explored them in depth. From our limited sample size of three, *NSNs at Risk* and *Bad Actors with Trend* appear to comprise roughly 1% of the total NSN population within their respective FSC codes, while *Bad Actors* constitute approximately 4% of total NSN population within their respective FSC codes.

We desire to expand upon this limited result and fully assess the impact of *NSNs at Risk*, *Bad Actors*, *and Bad Actors with Trend* as a function of quantity and cost. By executing our existing R script in a production loop, fixing the relative standard error to the same reasonable level for each regression tree, we extend the analysis to every FSC code in sufficient depth to comprehend the aggregate impact of items of concern to the U.S. Navy. Collectively, we find that *NSNs at Risk* and *Bad Actors with Trend* constitute approximately 1% in both U.S Navy consumable item population and annual consumable expenditure, and that *Bad Actors* comprise approximately 2% of U.S. Navy consumable item population and 7% of annual consumable expenditure (see Table 18).

Table 18. Consumable Items of Concern Summary Statistics, CY2015

Category	NSNs At Risk	Bad Actors	Bad Actors with Trend
U.S. Navy Consumable Population (unique NSNs)	268	6,128	657
U.S. Navy Consumable Population (%)	0.1%	2.0%	0.2%
Annual Consumable Expenditure ($, millions)	$3.8	$143.1	$19.4
Annual Consumable Expenditure (%)	0.2%	7.5%	1.0%
Total U.S. Navy Consumable Population (unique NSNs)	**300,281**		
Total Annual Consumable Expenditure ($, millions)	**$1,910**		

Consumable items of concern represent the collective group of *NSNs at Risk*, *Bad Actors*, and *Bad Actors with Trend*. We analyze over 300 unique FSC codes in the data in sufficient depth to obtain basic summary statistics on each category.

Although small in percentage of total consumable population and amount spent, all three categories of items of concern have a potentially large impact on Naval readiness and warrant further scrutiny.

V. CONCLUSIONS

A. SUMMARY

This thesis represents the first known attempt to formally define and classify consumable items of concern in the context of the U.S. Navy supply chain. Our proposed metric, customer time limit (CTL), normalizes requisitions ordered at different levels of priority from different regions of the world on the same time scale. The incorporation of CTL and coefficient of variation (CV) as metrics, in addition to statistical trends on the basis of regression tree model residuals, offers a robust method for classifying items of concern as either *NSNs at Risk*, *Bad Actors*, or *Bad Actors with Trend*.

When all FSC codes are collectively examined, we find that *NSNs at Risk* and *Bad Actors with Trend* constitute approximately 1% in both U.S Navy consumable item population and annual consumable expenditure ($19 million out of $1.9 billion purchased), and that *Bad Actors* comprise approximately 2% of U.S. Navy consumable item population and 7% of annual consumable expenditure ($140 million out of $1.9 billion purchased). In order to provide a better return for taxpayer dollars and improve Naval combat readiness, our classification system for U.S. Navy consumable items gives the Naval Supply Systems Command (NAVSUP) a better position for advocacy regarding these assets.

B. RECOMMENDATIONS

We offer three recommendations. First, replace average customer wait time (ACWT) with customer time limit (CTL) as the primary supply system metric for measuring responsiveness as a function of time. We have shown that it is a superior metric due to its ability to normalize requisitions ordered at different priority levels and locations around the world.

Second, examine each unique FSC code beyond the summary statistics level to refine the specific regression tree model as required, and continue to generate additional items of concern as future data becomes available. Our scripts are versatile and generic, and can be used to generate results from any FSC code using Inform-21 data.

Finally, we recommend that NAVSUP should use our results as a basis for a dialogue with DLA to improve the inventory position of the wholesale consumable inventory system. As explained in Chapter I, Section A, causes of shortages in the wholesale inventory the system may be attributed to one or more factors. However, the expanded use of two strategies identified in Chapter II, long-term contracts (LTCs) and collaborative forecasting, create a more agile and efficient consumable supply chain. Items with highly regular and frequent demand patterns should be identified for procurement under a LTC, which will nearly eliminate administrative lead time (ALT) in the contracting process. For other items, a collaborative forecast between NAVSUP or a major Navy customer and DLA should improve the quality of the forecast that DLA had previously been producing on its own.

C. FUTURE WORK

We recognize that the scope of our work is limited, and we welcome future research that builds upon our foundation. In order to strengthen the validity of the CTL metric, additional analysis could determine if the UMMIPS mandated order-to-receipt times are realistically scaled by geographic zone. Beyond CTL, other metrics, whether already in use or yet to be invented, may provide new insights in this research field. In addition, the specific criteria used in Table 6 for classifying items of concern is subjective and open to interpretation, producing either a more or less restrictive set of results.

Finally, we also recognize that the data used in our research come exclusively from the retail level of logistics and only considered Navy requisitions (Inform-21 database). This is a known shortcoming because items managed by DLA are requisitioned across military branches. Because our data cannot capture demand patterns beyond the Navy, important information that affects DLA inventory management remains hidden. Thus, we recommend that any future study incorporate wholesale inventory data from the Defense Logistics Agency's primary resource planning database, the Enterprise Business System (EBS) (GAO, 2014). This would allow a more robust analysis of wholesale inventory levels and trends, as well as specific insight as to the method, history, and challenges of procurement for each item.

APPENDIX A. PYTHON SCRIPT

```python
##IMPORT BLOCK##
import pandas as pd #Pandas is a data handling tool
import numpy as np #Numpy is a data handling tool
import datetime # Datetime is a date handling tool
import plotly.plotly as py #Python script to Plotly website API
import plotly.graph_objs as go #Python script to Plotly website API
###################

 #Author: LCDR Andrew Haley
#Project: Thesis 2016

#Script Purpose:
# 1) IMPORT EXTERNAL DATA FILES AND FILTER DATA PER TABLE 2 IN THESIS BODY
# 2) EXAMINE EVERY REQUISITION AND DETERMINE ITS MANDATED DELIVERY TIME; CALCULATE NEW CUSTOMER TIME LIMIT METRIC
# 3) LIMIT LEVELS OF CATEGORICAL VARIABLES (PROJECT CODE, SOURCE OF SUPPLY, POINT OF ENTRY, SERIES, AND REQUIRED DELIVERY DATE) TO THEIR MOST FREQUENT ENTRIES
# 4) DEVELOP COEFFICIENT OF VARIATION METRIC FOR EACH NSN IN DATA AND APPLY TO EACH REQUISITION
# 5) DEVELOP HEAT MAPS
# 6) DEVELOP HISTOGRAMS

############# PART 1: IMPORT EXTERNAL DATA FILES AND FILTER DATA PER TABLE 2 IN THESIS BODY############
##Import Data Files##
df2=pd.read_csv("birdtrackrawfile201320143B9B.csv",low_memory=False) #Read Raw Inform-21 File for 2013 and 2014
#df2=pd.read_csv("birdtrackrawfile20153B9B.csv",low_memory=False) #If 2nd time executing script, move comment symbol one row up and read Raw Inform-21 File for 2015
df_geocode=pd.read_csv("thesisuniquegeocodes.csv") #Read Geocode Priority File
df_ummips=pd.read_csv("UMMIPStable2.csv") #Read UMMIPS transportation timetable
dfAAC=pd.read_csv("AAC_TABLE_2016.csv") #Read reference AAC file from DLA that accurately categorizes NSNs as AAC = C, D, V, or Z
df_proj=pd.read_csv("ProjectCodeFreq20132014.csv") #Read in file with list of most frequent 50 Project codes
df_source=pd.read_csv("SourceSupplyCodeFreq20132014.csv") #Read in file with list of most frequent 50 Source of Supply codes
df_poe=pd.read_csv("POEFreq20132014.csv") #Read in file with list of most frequent 20 Point of Entry codes
df_series=pd.read_csv("SeriesFreq20132014.csv") #Read in file with list of most frequent 30 Series codes
df_rdd=pd.read_csv("RDDFreq20132014.csv") #Read in file with list of most frequent 30 Required Delivery Date codes
df_priorCV=pd.read_csv("20132014CV.csv") #Read in file that previously calculated the CV score for each NSN
######################
```

```
####Data Filtering per Table 2 in Thesis Body and Setting Column Types####
df2['FSC']= pd.to_numeric(df2['FSC'], errors='coerce') #turn NIINs numeric
df2=df2[(df2['CAT'] != 4) & (df2['CAT'] != 6) & (df2['CAT'] != 7) & #Excludes CAT 4 {Pending Stow}, 6 {Cancelled}, & 7 {Excluded}
(df2['FSC'] < 6500) & (df2['SOURCE_DOCID'].str.contains("A0|AT"))] #Excludes Weird FSC codes & Followup Requisitions
df2['Quantity']=abs(df2['Quantity']) #ensure positive values
df2['UP']=abs(df2['UP'])#ensure positive values
df2['Customer.Wait.Time']=abs(df2['Customer.Wait.Time'])#ensure positive values
df2['NIIN']= pd.to_numeric(df2['NIIN'], errors='coerce') #turn NSNs numeric
df2['Required.Delivery.Date']= pd.to_numeric(df2['Required.Delivery.Date'], errors='coerce') #Treats RDD codes as numbers
#df2 = df2.drop('Unnamed: 0', 1) #drop junk columns
df2 = df2.drop('SOURCE_DOCID', 1) #drop columns no longer needed
####################

###More Data Filtering: Ensure NSNs in Data Correspond to AAC Codes = C, D, V, or Z. Also, convert CAT variable to binary#####
#AAC codes are external to Inform-21 and requires querying DLA's EMALL website to obtain the accurate AAC code per NSN

#Ensure NSNs in data belong to AAC Codes = C, D, V, or Z
AACdict= {999999999999999: 'XXX'} #Create Initial Entry
for i in range(len(dfAAC)): #Loop to Convert AAC DataFrame to Dictionary
    AACdict[dfAAC.iloc[i,0]]=dfAAC.iloc[i,1]

myAAClist=[] #an empty list for later use
myBOOLlist=[] #an empty list for later use
myBB=[] #an empty list for later use
for i in range(len(df2)): #loop through each row in the dataset
    mytemprow=df2.iloc[i] #temporarily save each row
#use try/except construct of error catching with dictionaries;
#any NSN not found on reference list returns a non fatal-error and triggers "except" criteria
    try:
           myAAClist.append(AACdict[mytemprow['NIIN']]) #Desired AAC Codes are appended to a list
           myBOOLlist.append(True) #A Boolean list containing "TRUE" is created for rows with NSNs corresponding to desired AAC codes
    except KeyError: #the non-fatal error associated with incorrect dictionary entries; in other words, the NSNs that are not AAC codes = C,D, V, or Z
           myAAClist.append('ERROR') #in cases of error, list is appended "ERROR"
           myBOOLlist.append(False) #in cases of error, list is appended "FALSE"

#While this loop is primarily concerned with AAC codes, it is also a good opportunity to convert the CAT variable to binary
```

```
    if mytemprow['CAT'] == 2 or mytemprow['CAT'] == 5:
            myBB.append(1) #for those rows with a CAT=2 or 5 [means backorder] append 1
    else: #else append 0 (means no backorder)
            myBB.append(0)

df2.insert(len(df2.columns.values), 'AAC', myAAClist) #adds AAC Column to Data Frame
df2.insert(len(df2.columns.values), 'BB', myBB) #adds binary BB Column to Data Frame
df2 = df2.drop('CAT', 1) #drops original confusing CAT variable
df2=df2[myBOOLlist] #filter the dataframe down to only those rows with valid AAC codes
##############################################################################################


############# PART 2: EXAMINE EVERY REQUISITION AND DETERMINE ITS MANDATED DELIVERY TIME; CALCULATE NEW CTL METRIC#####
mymandatedtimelist=[] #an empty list to store the mandated delivery time per requisition
myCTLvaluelist=[] #an empty list to store the CTL time per requisition

for c in range(len(df2)): #loop through every row in dataframe

    dftemp_timetable=df2.iloc[c] #saves current row of dataframe temporarily
    geo=dftemp_timetable['GEOZONE_ORDERED'] #saves geographic code of current requisition
    p=dftemp_timetable['PRIORITY'] #saves priority code of current requisition
    ipg=dftemp_timetable['IPG'] #saves IPG code of current requisition

    myerrorcatch=df_geocode[df_geocode['GEOCODE']==geo] #ensures geo code saved matches to existing geo code reference table

    if pd.isnull(geo) or pd.isnull(p) or pd.isnull(ipg) or len(myerrorcatch)==0: #these requisitions fall short in some way; shortcircuiting the process to simply record a NaN score
            mymandatedtimelist.append(np.nan) #save the mandated time--as missing data
            myCTLvaluelist.append(np.nan) #save the excess time--as missing data
    else:
            rdd=dftemp_timetable['Required.Delivery.Date'] #saves RDD of requisition
            gw=dftemp_timetable['Series'] #saves Series code of requisition

 #the below section applies the UMMIPS timetable in P-485 Vol I Para 3049 to each requisition in the data set
    if p==1 or p==2 or p==3 and ipg == 1 and pd.notnull(rdd):
 #Conditions for TP1 material: Priority 1-3, IPG 1, and RDD filled in
 #Per P-485 Volume I para 3023 (page 3-23) high priority requisitions (IPG 1 & 2)
 #with a blank RDD will be automatically downgraded to IPG 3
            tpcode=1

    elif ipg == 2 and pd.notnull(rdd):
```

```
#Conditions for TP2 material: Pri 4-8 with RDD:[777] or Pri 4-15 with RDD:[444,555,777]
#However, since RDD is a manual entry field when ordering material and prone to frequent user error,
#chose to approximate the spirit of the publication by only selecting IPG 2 material with RDD filled in
        tpcode=2
    else:
#Conditions for TP3 material: Pri 4-15 with blank RDD or RDD eight days past requisition date
#Again, since RDD is prone to user error, tried to approximate this category with IPG 3 requisitions
#and those that failed the previous two categories for various reasons
        tpcode=3

    if gw == 'G' or gw == 'W':
#if requisition has G or W series (highest priority), ensure it is TP1 material and assign it as EXP category
        tpcode=1
        myrowcode='EXP' #EXP for express as defined in P485 Vol I para 3049
    else:
#for other requisitions, keep tpcode as determined in previous section and find appropriate
#alpha numeric row code based on geographic location at time requisition was placed
#the row codes are defined in P485 Vol I para 3049 as letters A-D, CONUS, and EXP
#each represent the mandated schedule for a particular part of the world given a particular priority level

#the row code is saved after being looked up in a pandas table
        myrowcode=df_geocode[df_geocode['GEOCODE'] == geo].iloc[0,tpcode]

        mymandatedtime=float(df_ummips[(df_ummips['TPAREA'] == myrowcode)].iloc[0,tpcode])
#mandated time is saved by using the row code just found above and the tp1 code in the previous section
#the value is looked up in a pandas table and saved as a variable

#Calculates the CTL Value per the Thesis Body definition
    myCTLvalue=(dftemp_timetable['Customer.Wait.Time']*100.0)/mymandatedtime

    if myCTLvalue<=0: #Ensures CTL values are at least 1
        myCTLvalue=1 #a CTL value less than or equal to zero would disrupt the planned natural logarithmic transformation for this variable

    mymandatedtimelist.append(mymandatedtime) #save the mandated time
    myCTLvaluelist.append(myCTLvalue) #save the CTL value

df2.insert(len(df2.columns.values), 'MandatedTime', mymandatedtimelist) #adds Mandated Time Column to dataframe
df2.insert(len(df2.columns.values), 'CustomerTimeLimit', myCTLvaluelist) #adds Customer Time Limit Column to dataframe
###############################################################################################
```

```
############ PART 3: LIMIT LEVELS OF CATEGORICAL VARIABLES TO THEIR MOST FREQUENT ENTRIES############
#Variables considered here are PROJECT CODE, SOURCE OF SUPPLY, POINT OF ENTRY, SERIES, AND REQUIRED DELIVERY DATE
#Each variable is generally limited to their top 20 to top 50 entries for calendar years 2013 to 2014
#Less frequent codes for these variables are simply labeled as "OTHER"

#Create Dummy Dictionaries with 1 entry
Projdict = {'999999999999999': '1'} #Create Initial Entry
Sourcedict = {'999999999999999': '1'} #Create Initial Entry
POEdict= {'999999999999999': '1'} #Create Initial Entry
Seriesdict= {'999999999999999': '1'} #Create Initial Entry
RDDdict= {999999999999999: 1} #Create Initial Entry

#Convert Reference DataFrames to Dictionaries for each variable
for i in range(len(df_proj)): #Loop to Convert Project Code and Source of Supply DataFrames to separate dictionaries
    Projdict[df_proj.iloc[i,0]]=df_proj.iloc[i,0]
    Sourcedict[df_source.iloc[i,0]]=df_source.iloc[i,0]

for i in range(len(df_series)): #Loop to Convert Series and Required Delivery Date DataFrames to separate dictionaries
    Seriesdict[df_series.iloc[i,0]]=df_series.iloc[i,0]
    RDDdict[df_rdd.iloc[i,0]]=df_rdd.iloc[i,0]

for i in range(len(df_poe)): #Loop to Convert Point of Entry DataFrame to a dictionary
    POEdict[df_poe.iloc[i,0]]=df_poe.iloc[i,0]

L11=[] #empty list for later use
L12=[] #empty list for later use
L13=[] #empty list for later use
L14=[] #empty list for later use
L15=[] #empty list for later use
for c in range(len(df2)): #cycle through entire dataframe
    bbb=df2.iloc[c]

#As with AAC section, use try/except error catching framework to determine entry in list
#see AAC section for line by line explanation
#if dictionary entry doesn't exist, instead of stopping program, it appends 'OTHER'
#efficiently accomplishes objectives without straining memory

    try:
        L11.append(Projdict[bbb['PROJECT.CODE']])
    except KeyError:
        L11.append('PROJ_OTHER')
```

```python

    try:
        L12.append(Sourcedict[bbb['SOURCE.OF.SUPPLY']])
    except KeyError:
        L12.append('SOS_OTHER')

    try:
        L13.append(POEdict[bbb['POE.RIC']])
    except KeyError:
        L13.append('POE_OTHER')

    try:
        L14.append(Seriesdict[bbb['Series']])
    except KeyError:
        L14.append('S_OTHER')

    try:
        L15.append(RDDdict[bbb['Required.Delivery.Date']])
    except KeyError:
        L15.append('RDD_OTHER')

df2.insert(len(df2.columns.values), 'PROJCODE', L11) #Add refined Project Code Variable to dataframe
df2.insert(len(df2.columns.values), 'SUPPLYSOURCE', L12) #Add refined Source of Supply Variable to dataframe
df2.insert(len(df2.columns.values), 'POE', L13) #Add refined Point of Entry Variable to dataframe
df2.insert(len(df2.columns.values), 'SeriesCode', L14) #Add refined Series Variable to dataframe
df2.insert(len(df2.columns.values), 'RDD', L15) #Add refined Required Delivery Date Variable to dataframe
df2 = df2.drop('Series', 1) #Drop old Series Variable with too many levels
df2 = df2.drop('PROJECT.CODE', 1) #Drop old Project Code Variable with too many levels
df2 = df2.drop('Required.Delivery.Date', 1) #Drop old Required Delivery Date Variable with too many levels
df2 = df2.drop('POE.RIC', 1) #Drop old Point of Entry Variable with too many levels
df2 = df2.drop('SOURCE.OF.SUPPLY', 1) #Drop Source of Supply
##########################################################################################


############# PART 4: DEVELOP COEFFICIENT OF VARIATION METRIC FOR EACH NSN IN DATA AND APPLY TO EACH REQUISITION#
#HERE YOU CAN EITHER RECYCLE THE CV VALUES GENERATED FROM A PRIOR RUN OF THIS SCRIPT (CHOICE A),
#OR GENERATE NEW CV VALUES FROM SCRATCH (CHOICE B)

#CHOICE A: RECYCLE EXISTING CV VALUES FROM 2013-2014 AND APPLY TO DATA (GENERATED IN PREVIOUS RUN OF THIS SCRIPT)
#Create Dummy Dictionaries with 1 entry
NIIN_CVdictA = {999999999999999: 1} #Create Initial Entry
NIIN_CVdictB = {999999999999999: 'NOT_HAPPENING'} #Create Initial Entry

```

```
for i in range(len(df_priorCV)): #Loop to Convert CV number score and CV categorical score to separate dictionaries
    NIIN_CVdictA[df_priorCV.iloc[i,0]]=df_priorCV.iloc[i,1]
    NIIN_CVdictB[df_priorCV.iloc[i,0]]=df_priorCV.iloc[i,2]

L25=[] #empty list for later use
L26=[] #empty list for later use
for z in range(len(df2)): #loop through each row of dataframe
    mytemprow=df2.iloc[z] #save current row temporarily

#Use try/except framework as before in AAC and Limiting Categorical levels sections; see AAC section for step by step guide
#for both of these, if the queried NSN is not contained in the reference list, it returns a non-fatal error which triggers appending "ERROR" to a list
    try:
        L25.append(NIIN_CVdictA[mytemprow['NIIN']])
    except KeyError:
        L25.append('CV_NUM_ERROR')

    try:
        L26.append(NIIN_CVdictB[mytemprow['NIIN']])
    except KeyError:
        L26.append('CV_CAT_ERROR')

df2.insert(len(df2.columns.values), 'CV_num', L25) #add new column in dataframe for CV numerical score
df2.insert(len(df2.columns.values), 'CV_cat', L26) #add new column in dataframe for CV categorical score
#CHOICE B: DEVELOP ORIGINAL CV SCORES BASED ON DATA
NIIN_CVdict = {999999999999999: 1}

groupedbyNIIN=df2.groupby(['NIIN']) #group everything by NSN; there will be about 300K unique NSNs from this data
dfNIINs=groupedbyNIIN['Quantity'].count() #creates count of NSNs in data
dfNIINs=dfNIINs.index #their index column is saved as a list

for z in dfNIINs: #loop through every NSN

    dftemp=groupedbyNIIN.get_group(z) #while looping, pulls each NSN out of original dataframe and saves that NSN group to a temp dataframe
    dfsize=len(dftemp) #saves number of requisitions in temp dataframe

    dfM = pd.DataFrame({ '0_MONTH' : pd.Categorical(['2013M1','2013M2','2013M3','2013M4','2013M5','2013M6','2013M7','2013M8','2013M9','2013M10','2013M11','2013M12','2014M1','2014M2','2014M3','2014M4','2014M5','2014M6','2014M7','2014M8','2014M9','2014M10','2014M11','2014M12']),
```

```
   'DEMANDCOUNT' : np.array([0] * 24)}) #Create Demand Pattern Summary Table for each month
#This second loop examines the quantity ordered in a given month and records the value to the "dfM" summary table; done for each NSN
        for d in range(len(dftemp)):
                zzz=dftemp.iloc[d]
#2013M1#
                if datetime.datetime.strptime('2013-01-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2013-01-31', "%Y-%m-%d"):
                        dfM.iloc[0,1]+=abs(zzz['Quantity'])
#2013M2
                elif datetime.datetime.strptime('2013-02-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2013-02-28', "%Y-%m-%d"):
                        dfM.iloc[1,1]+=abs(zzz['Quantity'])
#2013M3
                elif datetime.datetime.strptime('2013-03-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2013-03-31', "%Y-%m-%d"):
                        dfM.iloc[2,1]+=abs(zzz['Quantity'])
#2013M4
                elif datetime.datetime.strptime('2013-04-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d")<= datetime.datetime.strptime('2013-04-30', "%Y-%m-%d"):
                        dfM.iloc[3,1]+=abs(zzz['Quantity'])
#2013M5
                elif datetime.datetime.strptime('2013-05-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2013-05-31', "%Y-%m-%d"):
                        dfM.iloc[4,1]+=abs(zzz['Quantity'])
#2013M6
                elif datetime.datetime.strptime('2013-06-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2013-06-30', "%Y-%m-%d"):
                        dfM.iloc[5,1]+=abs(zzz['Quantity'])
#2013M7
                elif datetime.datetime.strptime('2013-07-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2013-07-31', "%Y-%m-%d"):
                        dfM.iloc[6,1]+=abs(zzz['Quantity'])
#2013M8
                elif datetime.datetime.strptime('2013-08-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2013-08-31', "%Y-%m-%d"):
                        dfM.iloc[7,1]+=abs(zzz['Quantity'])
#2013M9
                elif datetime.datetime.strptime('2013-09-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2013-09-30', "%Y-%m-%d"):
                        dfM.iloc[8,1]+=abs(zzz['Quantity'])
#2013M10
```

```
            elif datetime.datetime.strptime('2013-10-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2013-10-31', "%Y-%m-%d"):
                dfM.iloc[9,1]+=abs(zzz['Quantity'])
#2013M11
            elif datetime.datetime.strptime('2013-11-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2013-11-30', "%Y-%m-%d"):
                dfM.iloc[10,1]+=abs(zzz['Quantity'])
#2013M12
            elif datetime.datetime.strptime('2013-12-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2013-12-31', "%Y-%m-%d"):
                dfM.iloc[11,1]+=abs(zzz['Quantity'])
#2014M1#
            elif datetime.datetime.strptime('2014-01-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-01-31', "%Y-%m-%d"):
                dfM.iloc[12,1]+=abs(zzz['Quantity'])
#2014M2
            elif datetime.datetime.strptime('2014-02-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-02-28', "%Y-%m-%d"):
                dfM.iloc[13,1]+=abs(zzz['Quantity'])
#2014M3
            elif datetime.datetime.strptime('2014-03-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-03-31', "%Y-%m-%d"):
                dfM.iloc[14,1]+=abs(zzz['Quantity'])
#2014M4
            elif datetime.datetime.strptime('2014-04-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-04-30', "%Y-%m-%d"):
                dfM.iloc[15,1]+=abs(zzz['Quantity'])
#2014M5
            elif datetime.datetime.strptime('2014-05-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-05-31', "%Y-%m-%d"):
                dfM.iloc[16,1]+=abs(zzz['Quantity'])
#2014M6
            elif datetime.datetime.strptime('2014-06-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-06-30', "%Y-%m-%d"):
                dfM.iloc[17,1]+=abs(zzz['Quantity'])
#2014M7
            elif datetime.datetime.strptime('2014-07-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-07-31', "%Y-%m-%d"):
                dfM.iloc[18,1]+=abs(zzz['Quantity'])
#2014M8
            elif datetime.datetime.strptime('2014-08-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-08-31', "%Y-%m-%d"):
                dfM.iloc[19,1]+=abs(zzz['Quantity'])
```

```
#2014M9
                elif datetime.datetime.strptime('2014-09-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-09-30', "%Y-%m-%d"):
                        dfM.iloc[20,1]+=abs(zzz['Quantity'])
#2014M10
                elif datetime.datetime.strptime('2014-10-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-10-31', "%Y-%m-%d"):
                        dfM.iloc[21,1]+=abs(zzz['Quantity'])
#2014M11
                elif datetime.datetime.strptime('2014-11-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-11-30', "%Y-%m-%d"):
                        dfM.iloc[22,1]+=abs(zzz['Quantity'])
#2014M12
                elif datetime.datetime.strptime('2014-12-01', "%Y-%m-%d") <= datetime.datetime.strptime(zzz['DATE_ORDERED'], "%Y-%m-%d") <= datetime.datetime.strptime('2014-12-31', "%Y-%m-%d"):
                        dfM.iloc[23,1]+=abs(zzz['Quantity'])

    s_hat=dfM['DEMANDCOUNT'].std() #for each NSN, calculate sample standard deviation
    x_bar=dfM['DEMANDCOUNT'].mean() #for each NSN, calculate sample mean
    CV=s_hat/x_bar #for each NSN, calculate the CV score per the definition in the Thesis Body

    NIIN_CVdict[dftemp['NIIN'].iloc[0]] = CV #record the CV score per NSN in a dictionary

myCVlist=[] #an empty list for later use
for a in range(len(df2)): #loop through each row of dataframe
    yyy=df2.iloc[a] #save current row of dataframe temporarily
    myCVlist.append(NIIN_CVdict[yyy['NIIN']]) #assign CV score per row of dataframe using dictionary method

df2.insert(len(df2.columns.values), 'CV_num', myCVlist) #add new column for CV numerical score to dataframe

myCVcatlist=[] #an empty list for later use
for b in range(len(df2)): #loop through each row of dataframe
    xxx=df2.iloc[b] #save current row of dataframe temporarily
    tempvalCV=xxx['CV_num'] #assign a categorical value to the variable based on CV numeric score
    if tempvalCV<=1.0:
        myCVcatlist.append("CV_ULTRALOW")
    elif tempvalCV<2.0:
        myCVcatlist.append("CV_LOW")
    elif tempvalCV<3.4:
        myCVcatlist.append("CV_HIGH")
    else:
        myCVcatlist.append("CV_ULTRAHIGH")

```

```
df2.insert(len(df2.columns.values), 'CV_cat', myCVcatlist) #add new column for CV categorical score to dataframe
###########################################################################################


############# PART 5: DEVELOP HEAT MAP#############

#Part A: Calculate Median CTL for each grid location; Create Empty Grid to be filled later
groupedbyNIIN=df2.groupby(['NIIN']) #group everything by NSN
dfNIINs=groupedbyNIIN['Quantity'].count() #creates count of NSNs in data
dfNIINs=dfNIINs.index #their index column is saved as a list

### Create Empty Lists for Use Below###
L1=[]
L2=[]
L3=[]
L4=[]
L5=[]
L6=[]
L7=[]
L8=[]

for z in dfNIINs: #loop through every NSN

   dftemp=groupedbyNIIN.get_group(z) #while looping, pulls each NSN out of original dataframe and saves that NSN group to a temp dataframe
   dfsize=len(dftemp) #saves number of requisitions in temp dataframe

####### each list is appended with a different type of information about that NSN###
   L1.append(dftemp['FSC'].iloc[0])
   L2.append(z)
   L3.append(dftemp['COG'].iloc[0])
   L4.append(dfsize)
   L5.append(dftemp['Quantity'].sum())
   L6.append(dftemp['UP'].iloc[0])
   L7.append(dftemp['CustomerTimeLimit'].quantile(.5))
   L8.append(dftemp['CV_num'].iloc[0])


### Compiled lists are inserted into a pandas dataframe one column at a time####
df_results=pd.DataFrame(np.arange(len(L2)), columns=['test']) #a dummy column so that dataframe won't be empty at beginning
df_results.insert(len(df_results.columns.values), 'FSC', L1)
df_results.insert(len(df_results.columns.values), 'NIIN', L2)
```

```
df_results.insert(len(df_results.columns.values), 'COG', L3)
df_results.insert(len(df_results.columns.values), '#TimesOrdered', L4)
df_results.insert(len(df_results.columns.values), 'QtyOrdered', L5)
df_results.insert(len(df_results.columns.values), 'UP', L6)
df_results.insert(len(df_results.columns.values), 'CustomerTimeLimit_Median', L7)
df_results.insert(len(df_results.columns.values), 'CV_num', L8)
df_results = df_results.drop('test', 1) #drop the dummy column
##########
#create new Extended Money Value (EMV) column based on Quantity*Unit Price
df_results['TotalEMV']=abs(df_results['QtyOrdered'] * df_results['UP'])

mycostquantiles=[] #an empty list for storing cost quantiles
h=19 #starting value for the cost quantile variable
for z in range(19): # method to quickly generate a list of cost quantiles in decreasing order. Each entry is Q5% less than the previous one
   mycostquantiles.append(round(df_results['TotalEMV'].quantile(h/20.0),4)) #append the list with current entry
   h-=1 #decrement the cost quantile variable
   If h==0: #when h reaches zero,
          mycostquantiles.append(0) #append the list one last time with the value 0

myorderquantiles=[] #an empty list for storing order quantiles
g=19 #starting
for b in range(19): # method to generate a list of order quantiles in decreasing order. Each entry is Q5% less than the previous one
   myorderquantiles.append(df_results['#TimesOrdered'].quantile(g/20.0)) #append the list with current entry
   g-=1 #increment the cost quantile variable
   if g==0: #when h reaches zero,
          myorderquantiles.append(0) #append the list one last time with the value 0

lb=381 #a lower bound variable for use below
ub=401 #an upper bound variable for use below
mygrid=np.empty([0,20],dtype=np.int_) #an empty array for use below
mygrid=np.empty([0,20],dtype=np.int_) #an empty array for use below
for i in range(20): #method to quickly create a 20x20 grid filled with ascending numbers from bottom to top
   mygrid = np.vstack([mygrid,[np.arange(lb,ub)]]) #each row is created with a range of numbers: lower bound and upper bound
   lb=lb-20 #lower bound variable is decremented
   ub=ub-20 #upper bound variable is decremented

df_grid=pd.DataFrame(mygrid,columns=None) #stores the grid as a pandas dataframe

mycategorylist=[] #an empty list for storing the category label
```

```python
for i in range(len(df_results)): #loop through each row of dataframe and assign a numerical grid location based on Order Count Vs EMV
    zzz=df_results.iloc[i] #save current row to temp dataframe
    ooo=zzz['#TimesOrdered'] #pull frequency of order out and save as a variable
    ccc=zzz['TotalEMV'] #pull Extended Money Value out and save as a variable
    if pd.isnull(ooo) or pd.isnull(ccc): #in case data is missing, just add a 'NaN' entry
        mycategorylist.append(np.nan) #appends the list with 'NaN'

    else:
        yyy=19-myorderquantiles.index(next(x for x in myorderquantiles if x<=ooo)) #pulls index position for order frequency
        xxx=mycostquantiles.index(next(y for y in mycostquantiles if y<=ccc)) #pulls appropriate index position for EMV

    mycategorylist.append(df_grid.iloc[xxx,yyy]) #appends the list with the appropriate grid location

df_results.insert(7, 'GridLocation', mycategorylist) #insert new Column into results dataframe for Grid Location

mymedians=np.empty([0,2]) #create an empty array for later use
for v in range(400): #loop 400 times, same number as grid locations
    df_temp2=df_results[df_results['GridLocation'] == v+1] #loop through each grid location and pull out some stats
    mymedians = np.vstack([mymedians,[v+1, df_temp2['CustomerTimeLimit_Median'].quantile(0.5)]])

df_results_heat=pd.DataFrame(mymedians, columns=['GridLocation', 'CustomerTimeLimit_Median'])

#Part B: Create Visual Heat Map in Plotly
mymetric=df_results_heat['CustomerTimeLimit_Median']

lb=381 #a lower bound variable for use below
ub=401 #an upper bound variable for use below
myheatgrid=np.empty([0,20]) #an empty array for use below
for i in range(20): #method to quickly create a 20x20 grid filled with ascending numbers from bottom to top
    myrange=np.arange(lb,ub)
    mytempmetric=[]

for v in range(len(myrange)):
    mytempmetric.append(round(mymetric[(v+lb-1)],2))

    myheatgrid = np.vstack([myheatgrid,mytempmetric]) #each row is created with a range of numbers: lower bound and upper bound
    lb=lb-20 #lower bound variable is decremented
    ub=ub-20 #upper bound variable is decremented

```

```
dfheat=pd.DataFrame(myheatgrid,columns=np.arange(1,21))

for c in range(20): #this loop assembles the CTL scores into a form that can be read by the Plotly website
   columnnumber=c+1
   if pd.isnull(dfheat[columnnumber].sum()):
          dfheat.drop(columnnumber, axis=1, inplace=True)

g1=list(dfheat.iloc[19])
g2=list(dfheat.iloc[18])
g3=list(dfheat.iloc[17])
g4=list(dfheat.iloc[16])
g5=list(dfheat.iloc[15])
g6=list(dfheat.iloc[14])
g7=list(dfheat.iloc[13])
g8=list(dfheat.iloc[12])
g9=list(dfheat.iloc[11])
g10=list(dfheat.iloc[10])
g11=list(dfheat.iloc[9])
g12=list(dfheat.iloc[8])
g13=list(dfheat.iloc[7])
g14=list(dfheat.iloc[6])
g15=list(dfheat.iloc[5])
g16=list(dfheat.iloc[4])
g17=list(dfheat.iloc[3])
g18=list(dfheat.iloc[2])
g19=list(dfheat.iloc[1])
g20=list(dfheat.iloc[0])

myorderquantiles_unique=list(unique(myorderquantiles[:-1]))
mycostquantiles_sorted=sorted(mycostquantiles)

xvals=[] #creates categorical labels for the x-axis
for v in range(len(myorderquantiles_unique)):
   xvals.append(str(int(myorderquantiles_unique[v])))

yvals=[] #creates categorical labels for the y-axis
for v in range(len(mycostquantiles_sorted)):
   yvals.append("$"+str(int(round(mycostquantiles_sorted[v],0))))


#Finally, this section of code generates the Heat Map on the Plotly website.
#The heat colors have been manually developed to reflect a green-yellow-red continuum.
```

```
data = [
   go.Heatmap(
           z=[g1,g2,g3,g4,g5,g6,g7,g8,g9,g10,g11,g12,g13,g14,g15,g16,g17,g18,g19,g20],
           x=xvals,
           y=yvals,
           colorscale=[[0.0, 'rgb(0,246,0)'], [0.02, 'rgb(173,255,47)'], [0.07, 'rgb(255,255,0)'], [0.2,
'rgb(235,235,0)'], [0.3, 'rgb(216,216,0)'], [0.4, 'rgb(196,196,0)'],[0.5, 'rgb(243,139,0)'], [0.6, 'rgb(255,165,0)'],
[0.7, 'rgb(216,140,0)'], [0.8, 'rgb(177,114,0)'], [0.9, 'rgb(215,48,39)'], [0.95, 'rgb(165,0,38)'],
[0.97, 'rgb(126,0,29)'], [0.99, 'rgb(87,0,20)'], [1.0, 'rgb(47,0,11)']]
   )
]
plot_url = py.plot(data, filename='labelled-heatmap18')
###########################################################################################


############# PART 6: DEVELOP HISTOGRAMS############

#This section of code has the ability to generate a smart-looking histogram for any metric within any dataframe
#currently set to plot the histogram for entire data set, with Customer Time Limit as the metric of interest
data = [
   go.Histogram(
           x=df2['CustomerTimeLimit'],
   )
]
plot_url = py.plot(data, filename='basic-histogram17')

#print metric of interest summary statistics on screen
df2['CustomerTimeLimit'].describe()
###########################################################################################
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. R SCRIPT

```r
###LOAD LIBRARYS AND SET WORKING DIRECTORY###
library(doParallel)
registerDoParallel(cores=4)
library(rpart)
library(rpart.plot)
setwd("~/Documents/Thesis2016/DataWrangling")
###############
#Authors: LCDR Andrew Haley and Professor Robert Koyak
#Project: Thesis 2016
#SCRIPT PURPOSE:
# 1) READ FILES, SET COLUMN TYPES, AND PERFORM MISC PREPATORY WORK
# 2) BUILD RPART MODEL ON THE DATA WITHIN THE SPECIFIC FSC CODE CHOSEN
# 3) SAVE RESIDUALS FROM RPART TREE; PERFORM OTHER MISC WORK
# 4) QUANTCI FUNCTION (Conover, 1999)
# 5) BUILD ITEMS OF CONCERN DATAFRAMES
# 6) CREATE SCATTERPLOT OF RESIDUALS FOR SPECIFIC NSN
########## PART 1: READ FILES, SET COLUMN TYPES, AND PERFORM MISC PREPARATORY WORK##########
#Read Training File and Do Cleanup
Master20132014file_April2016 <- read.csv("~/Documents/Thesis2016/DataWrangling/Master20132014file_April2016.csv")
Master20132014file_April2016=Master20132014file_April2016[,-1] #Delete 1st Column, which is a leftover Python index column
Master20132014file_April2016$NIIN=as.factor(Master20132014file_April2016$NIIN) #set column type
Master20132014file_April2016$IPG=as.factor(Master20132014file_April2016$IPG) #set column type
Master20132014file_April2016$DATE_ORDERED=as.Date(Master20132014file_April2016$DATE_ORDERED) #set column type
Master20132014file_April2016$PRIORITY=as.factor(Master20132014file_April2016$PRIORITY) #set column type
Master20132014file_April2016$BB=as.factor(Master20132014file_April2016$BB) #set column type
Master20132014file_April2016$RDD=as.factor(Master20132014file_April2016$RDD) #set column type
Master20132014file_April2016$Quantity=abs(Master20132014file_April2016$Quantity) #ensure column contains positive values
Master20132014file_April2016$UP=abs(Master20132014file_April2016$UP) #ensure column contains positive values
Ntab = -sort(-table(Master20132014file_April2016$FSC)) #Create a ranking of requisitions found in data by FSC code
fscuniq = names(Ntab) #save FSC codes from data as a list for later use
#Read Test File and Do Cleanup
```

```
Master2015file_April2016 <- read.csv("~/Documents/Thesis2016/DataWrangling/Master2015file_April2016.csv")
Master2015file_April2016=Master2015file_April2016[,-1] #Delete 1st Column, which is a leftover Python index column
Master2015file_April2016$NIIN=as.factor(Master2015file_April2016$NIIN) #set column type
Master2015file_April2016$IPG=as.factor(Master2015file_April2016$IPG) #set column type
Master2015file_April2016$DATE_ORDERED=as.Date(Master2015file_April2016$DATE_ORDERED) #set column type
Master2015file_April2016$PRIORITY=as.factor(Master2015file_April2016$PRIORITY) #set column type
Master2015file_April2016$BB=as.factor(Master2015file_April2016$BB) #set column type
Master2015file_April2016$RDD=as.factor(Master2015file_April2016$RDD) #set column type
Master2015file_April2016$Quantity=abs(Master2015file_April2016$Quantity) #ensure column contains positive values
Master2015file_April2016$UP=abs(Master2015file_April2016$UP) #ensure column contains positive values

###GET TOTAL CONSUMABLE EXPENDITURE BY US NAVY FOR 2015
MYTOTAL2015EMV=data.frame(PRICE = Master2015file_April2016$UP, Q = Master2015file_April2016$Quantity)
MYTOTAL2015EMV=na.omit(MYTOTAL2015EMV)
MYtotal2015EMVvalue=with(MYTOTAL2015EMV,sum(PRICE*Q))
paste("The total US NAVY consumable expediture in CY2015 was $", MYtotal2015EMVvalue, sep = "")

#Remove Unnecessary Columns and Filter Data to a Specific FSC Code
removedcolumns=-c(1,3,9,14,22) #delete unneccesary columns

#FSC codes chosen in Thesis Body:
# 1. FSC code 5331 (position 3 in fscuniq list)
# 2. FSC code 4930 (position 75 in fscuniq list)
# 3. FSC code 1285 (position 153 in fscuniq list)

FSCcodechosen=fscuniq[144] #manually enter the position of the FSC code in the fscuniq list you want to analyze further

#This next section actually filters the rows and columns to only the FSC code chosen and dumps the junk columns
# na.omit command deletes any rows with missing data; missing data often creates errors in model building
reducedtrainingfile=Master20132014file_April2016[Master20132014file_April2016$FSC==FSCcodechosen,removedcolumns]
reducedtrainingfile=na.omit(reducedtrainingfile)

reducedtestfile=Master2015file_April2016[Master2015file_April2016$FSC==FSCcodechosen,removedcolumns]
reducedtestfile=na.omit(reducedtestfile)
#

#Extract some basic cost and population data about NSNs within this FSC code
uniquetrainingNSNs = -sort(-table(as.character(reducedtrainingfile$NIIN)))
summary(reducedtrainingfile$UP)
FSCamountpurchased_training=sum(reducedtrainingfile$UP*reducedtrainingfile$Quantity)
FSCrequisitioncount=dim.data.frame(reducedtrainingfile)[1]
```

```
########################################


########## PART 2: BUILD RPART MODEL ON THE DATA WITHIN THE SPECIFIC FSC CODE CHOSEN##########

#Create Tree on Training Set, with a low initial cp value
fsc.tree=rpart(log(CustomerTimeLimit)~.,reducedtrainingfile,cp=0.005)

printcp(fsc.tree) #print the relative error of the tree
plotcp(fsc.tree, col = 'blue') #plot the relative error of the tree
which.min(fsc.tree$cp[,4]) #print on screen row where error is minimized
fsc.tree.pruned=prune.rpart(fsc.tree, cp=0.03745) #manually enter new cp value to prune tree per previous relative error table and plot
printcp(fsc.tree.pruned) #print new pruned error tree relative error

mytreeprediction = predict(fsc.tree.pruned,newdata=reducedtestfile) #apply tree pred to testset data
########################################


########## PART 3: SAVE RESIDUALS FROM RPART TREE; PERFORM OTHER MISC WORK##########

answerset=reducedtestfile #save testset to a new dataframe
answerset$NIIN=as.character(answerset$NIIN) #set column type
answerset$Y=log(answerset$CustomerTimeLimit) #Add a Y column for the actual values
answerset$YHAT_RPART=mytreeprediction #Add a Y_HAT column for the predictions
answerset$RESID_RPART=answerset$Y-answerset$YHAT_RPART #their difference is the residual value
answerset$EMV=answerset$Quantity*answerset$UP #Add an EMV column just in case its needed
Ntab2 = -sort(-table(answerset$NIIN)) #create a ranking of NSNs in data for later use
niinuniq = names(Ntab2) #save NSNs in data to a list
########################################


########## PART 4: QUANTCI FUNCTION (Conover, 1999)##########

#Implements Conover's nonparametric 95% lower confidence bound (LCB)
#for the true population median consisting of the largest sample value

#Source:
#Conover, W. J. (1999). Practical Nonparametric Statistics (3rd ed.). New York: Wiley. ISBN: 978-0-471-16068-7

quantci = function(x, pval, CI = 0.95)
{
  n <- length(x)
```

```
  plo <- 0.5 * (1 - CI)
  phi <- CI + plo
  rlo <- qbinom(plo, n, pval)
  rlo <- rlo + (plo - pbinom(rlo, n, pval) + 1e-010 > 0)
  alo <- pbinom(rlo - 1, n, pval)
  shi <- qbinom(phi, n, pval)
  ahi <- pbinom(shi, n, pval)
  x <- sort(c( - Inf, x, Inf))
  return(cbind(p = pval, Lower = x[rlo + 1], Upper = x[shi + 2], Attained = ahi -
         alo))
}
#########################################

########## PART 5: BUILD ITEMS OF CONCERN DATAFRAMES##########
#Build a master dataframe of ampliflying information related to Spearman Test and Customer Time Limit (CTL) results per NSN
#Then apply the Classification Rules from Table 6 in Thesis Body to identify:
# NSNs at Risk, Bad Actors, and Bad Actors with Trend

n = length(niinuniq) #number of unique NSNs within this FSC code
mysize=round(n/2,0) #a size of dataframe to build; its current setting generally ensures at least 5 requisitions per NSN
X = data.frame(NSN = niinuniq[1:mysize], #build different column categories
Requisition.Count = numeric(mysize),Spearman.P.value = numeric(mysize),
CTL.Mean = numeric(mysize),SE.Mean = numeric(mysize),CTL.Median = numeric(mysize),
LCB_of_Median_CTL = numeric(mysize),
CV = factor(rep_len('CV_HIGH',mysize),levels=c('CV_HIGH','CV_LOW','CV_ULTRAHIGH','CV_ULTRALOW')))

for (j in 1:mysize) { #a loop to develop summary statistics per NSN relating to Spearman test, median/mean CTL, and LCB of Median CTL
  tt = answerset$NIIN == niinuniq[j]
  X[j,2] = sum(tt)
  X[j,3] = with(answerset[tt,],cor.test(as.numeric(DATE_ORDERED),RESID_RPART,method = "spearman",
  alternative = "greater",exact = F)$p.value) #this is the Spearman test; run for each NSN
  X[j,4] = mean(answerset[tt,11]) #Mean CTL value per NSN
  X[j,5] = sd(answerset[tt,11])/sqrt(sum(tt)) #standard error of the mean per NSN
  X[j,6] = median(answerset[tt,11]) #Median CTL value per NSN
  X[j,7] = quantci(answerset[tt,11],.5,.90)[2] #executes Conover's quantci function for the LCB of Median CTL per NSN
  X[j,8] = answerset[tt,18][1] #CV categorical score per NSN
}
for (j in 3:7) #round some of the columns to 5 decimal places
  X[,j] = round(X[,j],5)
```

```
#

#CREATE NSNS AT RISK DATAFRAME USING RULES FROM TABLE 6 IN THESIS BODY
NSNSATRISK_DF=X[(X$Spearman.P.value<=0.05) & (X$CV=='CV_LOW'|X$CV=='CV_ULTRALOW') & (X$LCB_of_Median_CTL>80 & X$LCB_of_Median_CTL<100),]

#CREATE BAD ACTOR DATAFRAME USING RULES FROM TABLE 6 IN THESIS BODY
BADACTOR_DF=X[(X$CV=='CV_LOW'|X$CV=='CV_ULTRALOW') & (X$LCB_of_Median_CTL>=100),]

#CREATE BAD ACTOR WITH TREND DATAFRAME USING RULES FROM TABLE 6 IN THESIS BODY
BADACTORWITHTREND_DF=X[(X$Spearman.P.value<=0.05) & (X$CV=='CV_LOW'|X$CV=='CV_ULTRALOW') & (X$LCB_of_Median_CTL>=100),]
#########################################


########## PART 6: CREATE SCATTERPLOT OF RESIDUALS FOR SPECIFIC NSN ##########
tt = answerset$NIIN == "12040634" #NSN from data to be plotted
with(answerset[tt,],plot(DATE_ORDERED,RESID_RPART,type = "p",pch = "*")) #create scatterplot
abline(h = 0,col = "red",lwd = 2.5)
with(answerset[tt,],cor.test(as.numeric(DATE_ORDERED),RESID_RPART,method = "spearman",
alternative = "greater")) #print Spearman test results for specific NSN on screen

#########################################
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. O-RING BAD ACTORS, COMPLETE TABLE, 2015

The items contained in this table have CV scores of less than 2 corresponding to variable values CV_ULTRALOW and CV_LOW.

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL	CV
5331-00-167-5122	346	1.000	214.3	214.3	CV_ULTRALOW
5331-00-165-1962	144	0.003	156.3	114.3	CV_ULTRALOW
5331-00-584-0263	131	0.998	125.0	100.0	CV_LOW
5331-01-133-9790	111	0.576	121.4	100.0	CV_LOW
5331-01-127-0971	88	0.003	125.0	100.0	CV_ULTRALOW
5331-00-248-3837	87	0.999	112.5	100.0	CV_ULTRALOW
5331-00-165-1970	80	0.741	178.6	106.3	CV_ULTRALOW
5331-01-006-2129	78	0.808	141.9	100.0	CV_LOW
5331-01-181-2509	67	0.035	150.0	100.0	CV_LOW
5331-01-587-8959	65	0.000	128.6	100.0	CV_LOW
5331-01-089-1583	65	0.998	200.0	171.4	CV_ULTRALOW
5331-00-542-1365	63	1.000	114.3	107.1	CV_LOW
5331-00-167-5141	59	0.702	128.6	114.3	CV_ULTRALOW
5331-01-113-5624	56	0.892	200.0	142.9	CV_LOW
5331-01-112-4060	55	0.978	157.1	100.0	CV_LOW
5331-01-129-7625	52	0.455	207.1	207.1	CV_LOW
5331-01-324-0916	50	0.982	142.9	142.9	CV_LOW
5331-00-580-4394	50	0.999	122.8	100.0	CV_LOW
5331-00-482-1595	50	0.645	171.4	152.9	CV_ULTRALOW
5331-00-480-8405	47	0.488	200.0	157.1	CV_LOW
5331-00-807-8993	46	0.880	196.9	142.9	CV_ULTRALOW
5331-01-097-2778	45	0.867	207.1	106.3	CV_LOW
5331-01-348-8331	45	0.232	125.0	100.0	CV_LOW
5331-00-157-6632	44	0.168	171.4	171.4	CV_LOW
5331-00-166-1020	44	0.815	182.9	100.0	CV_ULTRALOW
5331-00-501-9820	43	0.251	128.6	114.3	CV_LOW
5331-00-585-7487	43	0.060	142.9	114.3	CV_LOW
5331-00-833-1428	41	1.000	262.5	200.0	CV_LOW
5331-00-460-4674	40	1.000	219.6	168.8	CV_LOW
5331-01-007-1600	39	0.000	142.9	114.3	CV_LOW
5331-01-198-8439	39	1.000	657.1	257.1	CV_LOW

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL	CV
5331-00-480-4733	39	0.669	290.9	156.3	CV_ULTRALOW
5331-00-492-0575	38	0.947	189.3	114.3	CV_LOW
5331-01-015-6360	33	0.905	114.3	114.3	CV_LOW
5331-01-176-7915	33	0.993	242.9	118.8	CV_LOW
5331-00-410-4887	32	0.325	100.0	100.0	CV_LOW
5331-01-123-3302	29	0.387	200.0	100.0	CV_LOW
5331-01-094-5959	29	0.477	121.4	100.0	CV_ULTRALOW
5331-01-106-9574	28	0.069	142.9	131.3	CV_LOW
5331-01-183-0969	28	0.989	132.6	114.3	CV_LOW
5331-01-468-4214	28	0.334	171.4	136.2	CV_ULTRALOW
5331-00-579-7543	28	0.936	153.6	114.3	CV_ULTRALOW
5331-01-007-4895	27	0.953	285.7	142.9	CV_LOW
5331-01-460-9039	26	0.965	247.3	193.8	CV_ULTRALOW
5331-01-468-4209	25	0.001	185.7	136.2	CV_LOW
5331-01-433-1198	24	0.948	251.3	185.7	CV_LOW
5331-01-005-0534	23	0.617	671.4	371.4	CV_LOW
5331-01-051-5541	23	0.505	114.3	100.0	CV_LOW
5331-01-112-4059	23	0.581	142.9	114.3	CV_LOW
5331-01-164-3356	22	0.423	257.1	200.0	CV_LOW
5331-00-157-6630	22	1.000	271.4	200.0	CV_LOW
5331-00-419-0784	22	0.818	307.1	157.1	CV_LOW
5331-01-007-8595	21	0.611	114.3	100.0	CV_LOW
5331-01-065-7429	21	0.641	214.3	143.8	CV_LOW
5331-01-392-6718	21	0.611	121.4	100.0	CV_LOW
5331-00-127-2522	21	0.059	182.4	171.4	CV_LOW
5331-01-051-5540	20	0.656	371.4	171.4	CV_LOW
5331-01-147-4064	20	0.948	257.1	131.3	CV_LOW
5331-00-061-5471	20	0.766	228.6	200.0	CV_LOW
5331-00-579-8195	19	0.800	114.3	114.3	CV_LOW
5331-01-468-7846	18	0.767	136.2	120.5	CV_LOW
5331-00-166-1101	18	0.932	527.3	385.7	CV_LOW
5331-00-649-1911	18	0.363	200.0	200.0	CV_LOW
5331-01-019-2450	17	0.107	242.9	214.3	CV_LOW
5331-01-161-4498	17	0.475	131.3	107.1	CV_LOW
5331-01-207-9379	17	0.679	212.1	157.1	CV_LOW
5331-01-213-6763	17	0.677	114.3	100.0	CV_LOW

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL	CV
5331-01-351-2736	17	0.550	212.1	181.3	CV_LOW
5331-00-689-6480	17	0.033	285.7	200.0	CV_LOW
5331-00-817-7783	17	0.939	173.1	118.8	CV_ULTRALOW
5331-01-005-0521	16	0.496	163.8	118.2	CV_LOW
5331-01-005-0544	16	0.489	128.1	100.0	CV_LOW
5331-01-005-2305	16	0.044	214.3	153.8	CV_LOW
5331-01-091-1012	16	0.553	192.9	157.1	CV_LOW
5331-01-093-3503	16	0.004	220.5	156.3	CV_LOW
5331-01-343-2651	16	0.824	222.6	143.8	CV_LOW
5331-01-370-6912	16	0.125	220.5	175.0	CV_LOW
5331-01-419-3124	16	0.966	213.1	175.0	CV_LOW
5331-00-060-4325	16	0.170	192.9	122.7	CV_LOW
5331-00-291-3076	16	0.404	192.9	125.0	CV_LOW
5331-00-392-0762	16	0.598	167.9	142.9	CV_LOW
5331-01-138-7111	15	0.904	185.7	118.8	CV_LOW
5331-01-180-4801	15	0.777	142.9	100.0	CV_LOW
5331-00-252-6046	15	0.210	385.7	112.5	CV_LOW
5331-00-917-2612	15	0.798	171.4	107.1	CV_LOW
5331-01-029-3674	14	0.938	192.9	100.0	CV_LOW
5331-01-269-4323	14	0.217	171.4	142.9	CV_LOW
5331-00-126-5204	14	0.782	200.0	142.9	CV_LOW
5331-00-166-1092	14	0.263	109.4	100.0	CV_LOW
5331-00-701-1880	14	0.705	194.6	114.3	CV_LOW
5331-00-950-9715	14	0.632	242.9	142.9	CV_LOW
5331-01-121-0192	13	0.831	169.2	118.8	CV_LOW
5331-01-446-1185	13	0.188	328.6	138.5	CV_LOW
5331-00-753-1849	13	0.186	200.0	129.4	CV_LOW
5331-01-007-4899	12	0.682	514.3	242.9	CV_LOW
5331-01-108-3783	12	0.291	231.3	133.3	CV_LOW
5331-01-231-5217	12	0.018	192.9	171.4	CV_LOW
5331-01-393-5710	12	0.574	226.8	150.0	CV_LOW
5331-01-478-0043	12	0.992	421.8	106.3	CV_LOW
5331-01-113-2084	12	0.700	209.4	118.8	CV_ULTRALOW
5331-00-593-1247	12	0.743	399.6	193.8	CV_LOW
5331-01-024-2506	11	0.875	285.7	118.8	CV_LOW
5331-01-112-4058	11	0.440	142.9	115.5	CV_LOW

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL	CV
5331-01-130-7326	11	0.747	200.0	114.3	CV_LOW
5331-01-464-1400	11	0.984	300.0	142.9	CV_LOW
5331-00-172-7188	11	0.457	242.9	145.5	CV_LOW
5331-00-407-5727	11	0.652	228.6	142.9	CV_LOW
5331-00-763-2637	11	0.013	742.9	118.8	CV_LOW
5331-00-935-9150	11	0.877	123.1	100.0	CV_LOW
5331-00-285-9842	11	0.442	228.6	106.3	CV_ULTRALOW
5331-01-004-5034	10	0.758	200.0	114.3	CV_LOW
5331-01-021-1906	10	0.773	135.7	128.6	CV_LOW
5331-01-223-5505	10	0.923	142.9	119.0	CV_LOW
5331-01-225-4804	10	0.901	142.9	119.0	CV_LOW
5331-01-267-9176	10	0.500	314.3	135.7	CV_LOW
5331-01-005-3977	9	0.153	162.5	156.3	CV_LOW
5331-01-006-2110	9	0.552	200.0	128.6	CV_LOW
5331-00-061-2209	9	0.005	781.3	181.3	CV_LOW
5331-00-536-6835	9	0.999	771.4	271.4	CV_LOW
5331-01-031-8234	8	0.940	175.0	118.8	CV_LOW
5331-01-121-1714	8	0.993	157.1	114.3	CV_LOW
5331-01-206-6122	8	0.012	157.1	105.9	CV_LOW
5331-01-250-6735	8	0.000	107.1	107.1	CV_LOW
5331-01-092-2039	7	0.987	207.1	114.3	CV_LOW
5331-01-129-8896	7	0.263	287.5	129.5	CV_LOW
5331-01-169-3171	7	0.516	228.6	171.4	CV_LOW
5331-01-137-6897	7	0.560	218.8	145.5	CV_ULTRALOW
5331-00-472-3186	7	0.620	200.0	164.7	CV_LOW
5331-00-582-7665	7	0.928	257.1	150.0	CV_LOW
5331-00-840-6269	7	0.164	200.0	142.9	CV_LOW
5331-01-073-1219	6	0.432	196.9	100.0	CV_LOW
5331-01-081-3142	6	0.078	278.6	157.1	CV_LOW
5331-01-101-8014	6	0.827	200.0	143.8	CV_LOW
5331-01-160-4344	6	0.648	192.9	100.0	CV_LOW
5331-01-166-2100	6	0.075	228.6	228.6	CV_LOW
5331-01-213-5213	6	0.436	385.7	200.0	CV_LOW
5331-01-267-9175	6	0.717	342.9	135.7	CV_LOW
5331-01-285-1598	6	0.177	168.8	111.4	CV_LOW
5331-01-351-2739	6	0.979	221.4	114.3	CV_LOW

NSN	Requisition.Count	Spearman.P.value	Median CTL	LCB of Median CTL	CV
5331-00-400-7412	6	0.996	167.9	100.0	CV_LOW
5331-00-841-8564	6	0.939	257.1	200.0	CV_LOW
5331-01-031-8254	5	0.948	306.3	114.3	CV_LOW
5331-01-112-7959	5	0.374	300.0	157.1	CV_LOW
5331-01-169-2462	5	0.083	257.1	228.6	CV_LOW
5331-01-173-9224	5	0.370	200.0	142.9	CV_LOW
5331-12-184-9118	5	0.153	137.5	134.1	CV_LOW
5331-01-317-8092	5	0.467	200.0	125.0	CV_LOW
5331-01-360-0113	5	0.312	500.0	130.8	CV_LOW
5331-01-399-8395	5	0.026	171.4	114.3	CV_LOW
5331-01-416-7318	5	0.142	142.9	100.0	CV_LOW
5331-01-474-0024	5	0.944	285.7	257.1	CV_LOW
5331-01-034-3464	5	0.729	150.0	100.0	CV_ULTRALOW
5331-00-118-6330	5	0.161	314.3	200.0	CV_LOW
5331-00-406-5136	5	0.858	342.9	257.1	CV_LOW
5331-00-585-9473	5	0.086	171.4	142.9	CV_LOW
5331-00-753-1848	5	0.688	292.3	129.5	CV_LOW
5331-00-935-9203	5	0.891	228.6	200.0	CV_LOW

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Bratton, Michael, & Van De Walle, N. (1997). *Political regimes and regime transitions in Africa, 1910-1994*. (Study Number I06996). Ann Arbor: Inter-University Consortium for Political and Social Research.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth International Group.

Conover, W. J. (1999). *Practical nonparametric statistics* (3rd ed.). New York: Wiley. ISBN: 978-0-471-16068-7

Diaz, F. S., Cardenas, N., & Brito, L. O. (2006). *Implementing the National Inventory Management Strategy: A case study on DLA's National Inventory Management Strategy (NIMS)* (Master's thesis, Naval Postgraduate School, 2006). Monterey: Naval Postgraduate School. Retrieved from http://hdl.handle.net/10945/10066

Defense Logistics Agency. (2016). DLA at a glance. Retrieved April 2, 2016, from http://www.dla.mil/AtaGlance.aspx

Enthought Inc. (2016). Canopy (Version 1.6.2) [Computer software]. Retrieved from https://www.enthought.com/products/canopy/

Franklin, B. (1758). *The way to wealth*. ASIN: B005Z4G3H0

May, B. (2014). *Applying ABC Analysis to the Navy's Inventory Management System* (Master's thesis, Naval Postgraduate School, 2014). Monterey: Naval Postgraduate School. Retrieved from http://hdl.handle.net/10945/43953

Myers, J. L., & Well, A. (2003). *Research design and statistical analysis* (2nd ed.). Mahwah, NJ: Lawrence Erlbaum Associates. ISBN 0–8058–4037–0

Naval Supply Systems Command. (2015). *NAVSUP Publication 485: Operational Forces supply procedures* (5th ed., Vol. 1). Mechanicsburg, PA: Naval Supply Systems Command.

Naval Supply Systems Command. (2015). *NAVSUP Publication 485: Supply appendices* (5th ed., Vol. 2). Mechanicsburg, PA: Naval Supply Systems Command.

Peltz, E., Cox, A. G., Chan, E. W., Hart, G. E., Sommerhauser, D., Hawkins, C., & Connor, K. (2015). *Improving DLA supply chain agility* (Rep. No. RR-822-OSD). Santa Monica, CA: RAND Corporation.

R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from https://www.R-project.org/

Rigoni, M. P., & Correia de Souza, W. (2016). *Demand forecasting: An evaluation of DOD's accuracy metric and procedures* (Master's thesis, in press, Naval Postgraduate School, 2016). Monterey: Naval Postgraduate School.

United States Government Accountability Office Rep. No. GAO-14-495 (2014). Report to the Subcommittee on Readiness, Committee on Armed Services, House of Representatives.

Wackerly, D. D., Mendenhall, W., & Scheaffer, R. L. (2002). *Mathematical statistics with applications*. Pacific Grove, CA: Duxbury.

Therneau, Terry, Beth Atkinson, & Brian Ripley (2015). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-10. Retrieved from https://CRAN.R-project.org/package=rpart

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California